INTERACTIVE LEARNING AND ADAPTATION FOR
PERSONALIZED ROBOT-ASSISTED TRAINING

by

KONSTANTINOS TSIAKAS

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2018
To my closest friends and family for their unconditional support, encouragement and patience throughout these years.
ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my supervisor Prof. Fillia Makedon for her continuous support through all these years. I have been extremely lucky to have a supervisor who cares so much and supports my research. Besides my advisor, I would like to thank the rest of my thesis committee members: Prof. Manfred Huber Prof. Heng Huang, and Prof. Karkaletsis, for their insightful comments and guidance over the years.

More specifically, I would really like to thank Prof. Karkaletsis for providing me with this opportunity, through the joint Ph.D. program with National Center for Scientific Research – NCSR ”Demokritos”. My affiliation with NCSR provided me with valuable insights about my research, under the supervision and guidance of Prof. Karkaletsis, Maria Dagioglou and Stasinos Konstantopoulos. I would like to thank Prof. Huber, for both his excellent and inspiring course, which motivated and inspired me to formulate my research, and for his valuable comments and feedback. I would also like to thank all of my friends and labmates for the fruitfull collaboration and discussions we had over these years, as well as their emotional support!

March 9, 2018
ABSTRACT

INTERACTIVE LEARNING AND ADAPTATION FOR PERSONALIZED ROBOT-ASSISTED TRAINING

Konstantinos Tsiakas, Ph.D.

The University of Texas at Arlington, 2018

Supervising Professor: Fillia Makedon

Robot-Assisted Training (RAT) is a growing body of research in Human-Robot Interaction (HRI) that studies how robots can assist humans during a physical or cognitive training task. Robot-Assisted Training systems have a wide range of applications, varying from physical and/or social assistance in post-stroke rehabilitation to intervention and therapy for children with Autism Spectrum Disorders. The main goal of such systems is to provide a personalized and tailored session that matches user abilities and needs, by adjusting task-related parameters (e.g., task difficulty, robot behavior), in order to enhance the effects of the training session. Moreover, such systems need to adapt their training strategy based on user’s affective and cognitive states. Considering the sequential nature of human-robot interactions, Reinforcement Learning (RL) is an appropriate machine learning paradigm for solving sequential decision making problems with the potential to develop adaptive robots that adjust their behavior based on human abilities, preferences and needs. This research is motivated by the challenges that arise when different types of users are considered for real-time personalization using Reinforcement Learning, in a Robot-Assisted Training scenario.
To this end, we present an Interactive Learning and Adaptation Framework for Personalized Robot-Assisted Training. This framework utilizes Interactive RL (IRL) methods to facilitate the adaptation of the robot to each individual, monitoring both behavioral (task performance) and physiological data (task engagement). We discuss how task engagement can be integrated to the personalization mechanism, through Learning from Feedback. Moreover, we show how Human-in-the-Loop approaches can be used to utilize human expertise using informative control interfaces, towards a safe and tailored interaction. We illustrate this framework with a Socially Assistive Robotic (SAR) system that instructs and monitors a cognitive training task and adjusts task difficulty and robot behavior, in order to provide a personalized training session.

We present our data-driven approach (data collection, data analysis, user modeling and simulation), as well as a user study to evaluate our real-time SAR-based prototype system for personalized cognitive training. We discuss the limitations and challenges of our approach, as well as possible future directions, considering the different modules of the proposed system (RL-based personalization, user modeling, EEG analysis, Human-in-the-Loop). The long-term goal of this research is to develop personalized and co-adaptive human-robot interactive systems, where both agents (human, robot) adapt and learn from each other, in order to establish an efficient interaction.
# TABLE OF CONTENTS

ACKNOWLEDGEMENTS ................................................................. iv  
ABSTRACT ............................................................................... v  
LIST OF ILLUSTRATIONS ........................................................ x  
LIST OF TABLES ...................................................................... xiv  

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ROBOT-ASSISTED TRAINING: RECENT TRENDS AND NEEDS ......................... 1</td>
<td></td>
</tr>
<tr>
<td>1.1 Introduction to Robot-Assisted Training ........................................ 2</td>
<td></td>
</tr>
<tr>
<td>1.2 Related Taxonomies in HRI .......................................................... 3</td>
<td></td>
</tr>
<tr>
<td>1.3 A Taxonomy for Robot-Assisted Training ......................................... 7</td>
<td></td>
</tr>
<tr>
<td>1.3.1 Task Type and Requirements ................................................... 8</td>
<td></td>
</tr>
<tr>
<td>1.3.2 Interaction Types and Roles ................................................... 10</td>
<td></td>
</tr>
<tr>
<td>1.3.3 Level of Autonomy and Learning ............................................... 11</td>
<td></td>
</tr>
<tr>
<td>1.3.4 Personalization Dimensions ................................................... 13</td>
<td></td>
</tr>
<tr>
<td>1.4 Thesis Motivation and Outline .................................................... 14</td>
<td></td>
</tr>
<tr>
<td>2. REINFORCEMENT LEARNING FOR PERSONALIZATION ............................. 17</td>
<td></td>
</tr>
<tr>
<td>2.1 Personalization as Interaction Management problem .......................... 17</td>
<td></td>
</tr>
<tr>
<td>2.1.1 Preliminaries in Reinforcement Learning .................................... 19</td>
<td></td>
</tr>
<tr>
<td>2.1.2 Reinforcement Learning for Robot-Assisted Training ..................... 22</td>
<td></td>
</tr>
<tr>
<td>2.2 An RL-based Adaptive Rehabilitation Session Manager ..................... 25</td>
<td></td>
</tr>
<tr>
<td>2.2.1 Background and Related Work .................................................. 26</td>
<td></td>
</tr>
<tr>
<td>2.2.2 Approach and Methodology ...................................................... 27</td>
<td></td>
</tr>
<tr>
<td>2.2.3 Experimental Procedure and Discussion ...................................... 32</td>
<td></td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>6. TASK ENGAGEMENT AS PERSONALIZATION FEEDBACK</td>
<td>82</td>
</tr>
<tr>
<td>6.1 Using Task Engagement as Personalization Feedback</td>
<td>82</td>
</tr>
<tr>
<td>6.2 Learning Personalized Training Policies</td>
<td>84</td>
</tr>
<tr>
<td>6.2.1 User Simulation</td>
<td>85</td>
</tr>
<tr>
<td>6.2.2 Reinforcement Learning Experiments</td>
<td>86</td>
</tr>
<tr>
<td>6.3 Discussion and Future Work</td>
<td>89</td>
</tr>
<tr>
<td>7. USER STUDY AND FRAMEWORK EVALUATION</td>
<td>90</td>
</tr>
<tr>
<td>7.1 Study Protocol</td>
<td>90</td>
</tr>
<tr>
<td>7.2 Experimental Results and Discussion</td>
<td>93</td>
</tr>
<tr>
<td>8. CONCLUDING REMARKS AND FUTURE DIRECTIONS</td>
<td>97</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>104</td>
</tr>
<tr>
<td>BIOGRAPHICAL STATEMENT</td>
<td>122</td>
</tr>
</tbody>
</table>
# LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Taxonomy Categories for Robot-Assisted Training</td>
<td>7</td>
</tr>
<tr>
<td>1.2</td>
<td>Interaction Types in Robot-Assisted Training (inspired by [1])</td>
<td>11</td>
</tr>
<tr>
<td>2.1</td>
<td>The interaction loop with an interactive learning agent</td>
<td>17</td>
</tr>
<tr>
<td>2.2</td>
<td>The Reinforcement Learning setup</td>
<td>19</td>
</tr>
<tr>
<td>2.3</td>
<td>Model-free algorithms: Q-learning (off-policy) and SARSA (on-policy)</td>
<td>21</td>
</tr>
<tr>
<td>2.4</td>
<td>Robot-Assisted Training as a Reinforcement Learning problem</td>
<td>22</td>
</tr>
<tr>
<td>2.5</td>
<td>A telerehabilitation setup. A virtual agent acts as an exercise coach.</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>The patient performs the prescribed exercise and the system monitors user performance.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>Courtesy of Swedish Health Care, RoboBusiness Leadership Summit, 2012</em></td>
<td></td>
</tr>
<tr>
<td>2.6</td>
<td>The proposed Dialogue System for Safe Rehabilitation.</td>
<td>28</td>
</tr>
<tr>
<td>2.7</td>
<td>ReADAPT manager formulated as a Reinforcement Learning problem.</td>
<td>29</td>
</tr>
<tr>
<td>2.8</td>
<td>Simulated user models</td>
<td>32</td>
</tr>
<tr>
<td>2.9</td>
<td>Screenshot of the ReADAPT user interface.</td>
<td>36</td>
</tr>
<tr>
<td>3.1</td>
<td>The training task formulated as a Reinforcement Learning problem.</td>
<td>40</td>
</tr>
<tr>
<td>3.2</td>
<td>User model examples. Task performance based on task difficulty and duration.</td>
<td>41</td>
</tr>
<tr>
<td>3.3</td>
<td>Learning experiments. Applying Q-learning for the different user models results to different user-specific policies (USP).</td>
<td>42</td>
</tr>
<tr>
<td>3.4</td>
<td>Policy Transfer experiments. In this experiment, we applied all learned USP to the different user models.</td>
<td>43</td>
</tr>
</tbody>
</table>
3.5 Interactive Reinforcement Learning approaches

3.6 We extend the RL framework by adding two additional communication channels; feedback and guidance. Their integration to the adaptation module can enable the agent to continuously adapt towards the current user, ensuring a safe and personalized interaction.

3.7 Integrating feedback and guidance to facilitate policy adaptation

4.1 Flow Theory [2] and Yerkes-Dodson Law [3]. Relation between user performance and skills, task difficulty and affective states (engagement, anxiety, etc.)

4.2 Muse headband and electrode locations by 10-20 international standard

4.3 The Sequence Learning task setup

4.4 Survey Results during data collection

4.5 Task performance and engagement for different users

4.6 User clustering using multidimensional scaling and K-means (left), cluster means as success probabilities at each level (middle) and mean engagement per level (right).

4.7 System Architecture for Real-Time Personalization

4.8 System Procedure for Personalization

4.9 System Prototype for Real-Time Personalization

5.1 Experimental Protocol

5.2 The participant interacts with three different interfaces during a simulated robot assisted training assessment session. The (a) control-only interface does not have any monitoring features. The (b) history-based GUI provides a history of task performance over past levels, while (c) the model-based interface provides a visualization of the performance model as a set of success probabilities at each level.
5.3 The participant completes a session survey after each assessment session

5.4 Participants' feedback on the enjoyability and the effectiveness of the three interfaces

5.5 The mean squared errors between the survey data and both the actual data and the pre-calculated user models

5.6 A GUI prototype for Intelligent Monitoring and Control

6.1 The Sequence Learning task as an RL problem

6.2 Reinforcement Learning setup using simulated users

6.3 Q-Learning results for the different user models. We visualize task performance and task engagement for each user model. Task engagement as personalization feedback can facilitate learning

6.4 Visualization of the learned policies for a given state: $\pi(state) = P(action|S_0)$. The x-axis shows the possible actions and y-axis shows the probability for each action. Each row corresponds to a user model and each column corresponds to the learning method

7.1 Study Protocol for System Evaluation

7.2 Final Study Survey

7.3 Task Performance and User Survey Results

7.4 RMS distances for user models and policies before and after the personalized training session

8.1 HCPS system architecture

8.2 Future and ongoing work on developing an experimental testbed for vocational cognitive assessment and training

8.3 Vocational Assessment and Training
8.4 Cognitive and emotion models can be utilized to inform personalized robot-assisted assessment and training systems. Moreover, such systems can be used as experimental testbeds to study existing cognitive and affective architectures.
LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>An Updated Taxonomy in HRI [1]</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>Considering our proposed taxonomy, we classify recent works in Robot-Assisted Training based on (a) Task Type and Requirements, (b) Interaction Types and Roles, (c) Level of Autonomy and Learning and (d) Personalization Dimensions</td>
<td>9</td>
</tr>
<tr>
<td>1.3</td>
<td>Personalization Dimensions. A list of control and observed parameters for personalization</td>
<td>14</td>
</tr>
<tr>
<td>2.1</td>
<td>ReADAPT as a Markov Decision Process</td>
<td>30</td>
</tr>
<tr>
<td>3.1</td>
<td>Interactive Reinforcement Learning Approaches</td>
<td>46</td>
</tr>
<tr>
<td>4.1</td>
<td>Examples of encouraging and challenging feedback</td>
<td>59</td>
</tr>
<tr>
<td>6.1</td>
<td>The defined MDP of the problem</td>
<td>84</td>
</tr>
</tbody>
</table>
CHAPTER 1

ROBOT-ASSISTED TRAINING: RECENT TRENDS AND NEEDS

The motivation of this research is personalization in Human-Robot Interaction (HRI), focusing on the different roles of human users who participate in the interaction, as well as the different types of communication signals and feedback, based on which human users and robots interact. Personalization is intended to facilitate and enhance the interaction between users and robots and is integral to meeting the system requirements and the goals for which an HRI system is designed. For example, personalization of social robots in health settings should not only address the needs of the users (patients, caregivers), but also allow these different types of users to actively participate in this personalization procedure [4]. Personalization is essential for ensuring safety in shared-environment human-robot collaboration settings. Collaborative robots should be able to adapt and personalize their strategies based on human preferences, task proficiency and expertise, affective and cognitive states [5].

Despite the effectiveness of HRI in several application areas, personalization remains an open issue in HRI research and depends on various challenges, such as modeling user’s preferences, skills, intentions and emotions, as well as keeping track of previous interactions in a real-time environment [6]. Moreover, human supervisors, who are not technical experts (e.g., therapists), should also participate actively in personalization by directly adjusting robot parameters (e.g., robot personality) through human-robot graphical user interfaces [7]. Robots can also be used as interactive learning systems that utilize users assistance to incrementally improve robots perception skills [8, 9].
This thesis focuses on how different types of users (primary user, supervisor) and feedback (implicit, explicit), as well as prior experience and past interactions can be used for personalization, in the context of a real-time personalized Robot-Assisted Training system.

1.1 Introduction to Robot-Assisted Training

Robot-Assisted Training (RAT) is a growing body of research in Human-Robot Interaction (HRI) that studies how robots can assist humans in a task-dependent interaction. RAT systems have a wide range of applications, varying from physical assistance in post-stroke rehabilitation and robotic prosthetics [10, 11], cognitive training for patients suffering from dementia and Alzheimer’s disease [12, 13], to intervention and therapy for children with Autism Spectrum Disorders [14, 15, 16] and tutoring systems for language learning and children education [17, 18, 19]. As a multidisciplinary research field, it requires expertise in several research areas, including robotics, human-machine interaction, machine learning, data mining, computer vision, as well as psychology and educational sciences, kinesiology, occupational therapy and others. Despite this large variety of applications, target populations and system requirements, a common goal of Robot-Assisted Training systems is to enhance user’s performance by providing safe, personalized and targeted assistance towards maximizing training effects. Personalization has the potential to create a tailored and compelling experience that encourages and assists users to perform a given task and meet the training goals.

In order to address the needs of personalization in the context of a Robot-Assisted Training system, our first step is to identify a common set of parameters (i.e., taxonomy categories) that characterize a Robot-Assisted Training system [petra18], taking into consideration related taxonomies in HRI, as well as recent works
in Robot-Assisted Training systems, in order to highlight the current research trends and challenges in this growing HRI area.

1.2 Related Taxonomies in HRI

One of the most generalized and broad classifications for HRI systems provides a classification framework based on eleven taxonomy categories [20, 1]: task type, task criticality, robot morphology, ratio of people to robots, composition of robot teams, level of shared interaction among teams, interaction roles, physical proximity, decision support for operators, time-space taxonomy and autonomy level/amount of interventions from operators.

<table>
<thead>
<tr>
<th>System Requirements</th>
<th>Interaction Type</th>
<th>Human Roles</th>
<th>Spatio-temporal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Type</td>
<td>Ratio of People to Robots</td>
<td>Human Interaction Roles</td>
<td>Time-Space Taxonomy</td>
</tr>
<tr>
<td>Task Criticality</td>
<td>Level of Shared Interaction Among Teams</td>
<td>Decision Support for Operators</td>
<td>Human-Robot Physical Proximity</td>
</tr>
<tr>
<td>Robot Morphology</td>
<td>Composition of Robot Teams</td>
<td>Level of Autonomy-Amount of Intervention</td>
<td></td>
</tr>
</tbody>
</table>

Table 1.1. An Updated Taxonomy in HRI [1]

These different categories can be used to characterize and classify an HRI system. **System requirements** can be defined by the task type, task criticality and robot morphology. Task type refers to task description and representation in a high-level way. It is important because it sets the system requirements and the basic design guidelines. Some possible values of this variable are: tutoring system, collaborative manufacturing training, rehabilitation exercises, etc. Task criticality considers safety issues (e.g., human safety risk) and has three possible values: high, medium, low. Since robot appearance affects how people interact with it, robot morphology vari-
able describes the robot appearance type, i.e., anthropomorphic, zoomorphic, and functional.

Depending on the application, there are different interaction types and roles for human and robot members. One parameter under this category is the ratio of people to robots, which simply defines the number of humans over robots participating in the interaction. Another parameter is the type of interaction between human and robot participants, defining the level of shared interaction among (robot and human) teams (as we show in Figure 1.2). The most straightforward example is a single robot that interacts with a single human user. A more complex example is a human operator that sends commands to a team of robots, which has to autonomously coordinate its members to execute the command. Another example is a team of human users that coordinates and sends specific commands to independent robots. Different team formulations and member roles require different interaction protocols. Another parameter under this category, which can affect the interaction protocols, is composition of robot teams, considering different robot types.

Since human participation is essential for any HRI system, human roles must be well-defined. Scholtz [21] has defined five different roles for a human participant in an interaction with a robot: supervisor, operator, teammate, mechanic/programmer and bystander. Moreover, two more are added by Goodrich [22]: mentor and information consumer. In many applications, where the human acts as an operator or supervisor, an HRI system should provide the user with decision support. The human user needs to monitor, intervene, and modify robot’s behavior, when needed. Providing appropriate information to the operator can enhance their decision making. For example, the robot can visualize information about the list of all available sensors and data streams. Interactive methods can be used to make the system’s decision process transparent to the user, as humans and machines require shared awareness
and shared intent during human-robot interactions [23, 24]. Another defining factor for HRI is the level of autonomy (or the amount of human intervention). Human operators or supervisors often have the ability to control the robot and modify its behavior. The level of the autonomy is defined as the amount of time that the robot acts in an autonomous manner. In many cases, this value can be adjusted during the interaction, resulting to a progressively autonomous system. Human workload and cognitive capacity are two important factors to take into consideration in order to define the level of autonomy.

Other parameters that are defined by this taxonomy are spatiotemporal and define human-robot interaction in terms of space and time. More specifically, this parameters categorize an HRI system based on whether human and robot share the same space (collocated, non-collocated), and whether they act at the same time or not (synchronous, asynchronous). Moreover, in a collocated HRI system, the robot can be defined by different proximity behaviors e.g., avoiding, passing, following, approaching, touching, and/or none. Focusing on specific applications and domains requires a more detailed description. For example, SAR systems have been used for physical rehabilitation [25], where proxemics are defined based on social interaction zones (e.g., social, personal, intimate) used to define robot’s personality (e.g., introvert, extrovert).

Depending on the application and the system requirements, several taxonomies have been introduced for human-robot interaction systems, such as human-robot collaboration, child-robot interaction, assistive robotics and others. More specifically, Salter [26] presented a taxonomy for child-robot interaction (CRI), based on the control factors for both robots and participants. They used three categories for both robots and human participants: Autonomy, Group and Environment. For example, the robotic autonomy (RA) can be classified as one of the following: autonomous,
fixed, combination, Wizard of Oz (WoZ), and remote-controlled. The participant autonomy (PA) can be: free, natural, comfortable, directed, and controlled, based on how users are allowed to interact with the robot. The authors have provided a taxonomy rating in relation to participant and robot influences, for all three categories. They used a rating scale from 1 (None) to 9 (High) to describe the level of control for robots and participants.

Other taxonomies focus and elaborate on specific parameters, such as robot autonomy level. In [27], the authors present a framework for Levels Of Robot Autonomy (LORA) in HRI, identifying parameters that influence and get influenced by level of robot autonomy. They provided a flow chart as a guideline to determine and define robot autonomy and its effects on HRI. Their taxonomy for robot autonomy takes into consideration the level of autonomy during sensing, planning and acting. The guidelines can be used to identify task and environmental influences on robot autonomy level, measure and categorize autonomy level and identify HRI parameters that have an impact on robot autonomy.

Focusing on human-robot collaboration systems, another recent taxonomy describes the level of automation, focusing on collaborative robots [28]. The Interaction Readiness Model (IRM) classifies a system in one of the four levels, based on the level of automation. This model correlates the level of automation with task complexity in a manufacturing environment. The automation level varies from gated-robots mode, where robot is idle while human is present, to fully interactive mode, where humans and robots learn how to solve a synergistic task in a collaborative manner. This model has been defined based on real industrial needs, towards Industry 4.0 and ”robofacturing” [29].
1.3 A Taxonomy for Robot-Assisted Training

Based on the existing taxonomies and classification frameworks, we propose a list of parameters that need to be considered for the design and development of a Robot-Assisted Training system. The defined categories are: **Task Type and Requirements**, **Interaction Types and Roles**, **Level of Autonomy and Learning** and **Personalization Dimensions**, as we show in Figure 1.1. Considering these categories, we review recent works in order to highlight the current trends and research challenges, as well as the relationship between them, e.g., the requirements of a rehabilitation system (high level of task criticality) may require a supervisor to monitor the interaction - *interaction roles*. An indicative set of examples is shown in Table 1.2.

![Figure 1.1. Taxonomy Categories for Robot-Assisted Training.](image)
1.3.1 Task Type and Requirements

When designing a Robot-Assisted Training system, the task type and requirements are the first parameters to be defined, since they can set the tone for the overall design, implementation and evaluation process. The task type and requirements also define other important parameters as task criticality and safety issues, target populations, robot morphology and type of assistance. The task type must provide a high-level description of the task and the system environment. This category can also include parameters such as type of assistance, set of appropriate sensors, level of obtrusiveness, and others. Based on a recent taxonomy [30], types of assistive robots (based on type of assistance) include physically assistive robotics (PAR), socially assistive robotics (SAR), as well as sensory and feedback systems.

In a recent work [31], the authors have presented a SAR-based system for language learning with children using a tablet learning game. The proposed system uses a camera to capture and analyze facial expressions and affective features (gaze, smile, engagement, valence, etc), during the interaction. The robot uses this information in order to provide a personalized affective policy by adjusting its social verbal behavior (valence and engagement of spoken instructions) and keep the child in a positive affective state. Another work presents a physically assistive robot for upper-limb rehabilitation [32]. In this work, the authors presented an automated system for a rehabilitation robotic (physically assistive) device that guides stroke patients during an upper-limb reaching task. The system uses task-related observation (e.g., task completion time and assistance needed) to estimate user-related metrics (e.g., user fatigue, progress, etc.) and adapt the reaching task parameters to enhance training effects. As part of the system’s requirements, the authors argue that the use of sensors (camera, electromyography- EMG sensors, etc.) could lead to noisy and untrustwor-
due to high task criticality, a supervisor monitors the system’s decisions and intervenes when needed.

<table>
<thead>
<tr>
<th>Task Type and Requirements</th>
<th>Interaction Types and Roles</th>
<th>Level of Autonomy and Learning</th>
<th>Personalization Dimension(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Socially Assistive Robotics (SAR) for Language Learning with Children [31]</td>
<td>A social robot acts as an affective tutor during a language learning game</td>
<td>The robot acts fully autonomously and learns using Reinforcement Learning (RL)</td>
<td>The robot adjusts its engagement and valence during verbal instructions</td>
</tr>
<tr>
<td>SAR-based system for Post Stroke Rehabilitation for Elderly Patients [25]</td>
<td>The robot therapist monitors, assists, encourages users during rehabilitation</td>
<td>The robot acts fully autonomously and personalizes its policy using PGRL</td>
<td>The robot adjusts its therapy style, speed and proxemics based on user progress</td>
</tr>
<tr>
<td>Robot-Based Rehabilitation using Serious Games and Haptic device [33]</td>
<td>The user performs a reaching task using a robotic haptic device</td>
<td>The robot acts autonomously and learns through RL</td>
<td>The system adjusts the game parameters to challenge the user</td>
</tr>
<tr>
<td>Adaptive Upper-Limb Rehabilitation using a Robotic Haptic Device [32]</td>
<td>The robotic arm trains the user in a reaching task. A supervisor monitors system’s decisions</td>
<td>The robot acts autonomously based on a given policy (no learning); an expert can alter the action</td>
<td>The system decides reaching target, resistance level of resistance, or when the task should stop</td>
</tr>
<tr>
<td>Social Robot for Attention Acquisition during a Memory Game [34]</td>
<td>The robot acts as a tutor who guides user’s attention during a memory game, in a WoZ setup</td>
<td>The system acts semi-autonomously. A supervisor provides RL with user state to select gestures</td>
<td>The robot learns the appropriate gesture combination to increase user attention</td>
</tr>
<tr>
<td>Physical Exercising for Children using a Social Robot and Wizard-of-Oz Interfaces [35]</td>
<td>The robot shows the exercises to be performed. A supervisor can control the robot</td>
<td>The system acts in a semi-autonomous manner. The robot learns from human input</td>
<td>The robot personalizes the exercise regimen according to exercise performance and compliance</td>
</tr>
</tbody>
</table>

Table 1.2. Considering our proposed taxonomy, we classify recent works in Robot-Assisted Training based on (a) Task Type and Requirements, (b) Interaction Types and Roles, (c) Level of Autonomy and Learning and (d) Personalization Dimensions

Socially assistive robots can provide supportive behavior, tailored instructions and recommendations, as well as attention acquisition to assist users in several appli-
cations, e.g., retain attention through gestures in a memory game [34]. Another example demonstrates how socially assistive robots can be deployed for physical rehabilitation [25], investigating different robot behavior parameters (human-robot personality matching, robot proxemics, etc.). Social assistance can also improve compliance and performance for physical exercising in child-human interaction [35].

1.3.2 Interaction Types and Roles

Similar to previous taxonomies, we define the human-robot interaction types and roles. These parameters define the interaction types; how the human-robot team is formulated and communicates, as well as the interaction roles for each member of the interaction.

As described by the HRI taxonomy [1], human users and robots can interact with different ways, formulating different types of teams, according to the system requirements and the application. In Figure 1.2, we show the different interaction types of humans and robots in a Robot-Assisted Training system. For example, a single robot can interact either independently with two different users (Figure 1.3.B) or with a team of users (Figure 1.3.C). In the first case, a robot may supervise a RAT sessions or multiple RAT sessions (Figure 1.3.E), while in the latter case, human members need to coordinate with each other in order to execute the task (e.g., multi-party collaborative assembly task).

Considering interaction roles, previous taxonomies have focused on human members roles [20, 1], as mentioned in Section 1.2. In this work, we present interaction roles that both human and robot members can have in a Robot-Assisted Training session, including: primary user, supervisor, operator, instructor and teammate. A primary user is the end user who participates actively in the interaction (e.g., patient). While the most frequent case is that this is a human user, there are works
that focus on training a secondary user (therapist) [36], or even simulate the primary user using the robot in order to evaluate the system from the aspect of the supervisor [37]. A (human or robot) supervisor monitors the training session (i.e., through sensors or interfaces) to capture essential information of the training session (e.g., task parameters, user performance and condition, etc.). A human operator can control the parameters of the training session (e.g., using a control interface), while a robot operator is the actual actuator which assists the user in the task (e.g., robotic arm for wrist rehabilitation [33]). An instructor plays the role of a tutor who guides and instructs the user during the task (e.g., educational robots). Team co-ordination and collaboration can be used as training tasks, thus the role of a (human or robot) teammate who interacts with the user, can be an important member role in a training session.

1.3.3 Level of Autonomy and Learning

An essential aspect of a Robot-Assisted Training system is the level of robot autonomy, which defines whether the robot acts autonomously or under the guidance of
a human user. Specific system requirements and parameters may require the presence of a human expert who acts as a supervisor to ensure safety and efficiency during the training session. Influenced by LORA [27], the level of autonomy in a RAT system varies from tele-operation to fully-autonomous systems, including sliding and shared autonomy systems [38, 39], where human experts and robot can work together to achieve shared goals.

For example, in the upper-limb reaching task example (Section 3.1), the system suggests an action to the supervisor, through a GUI, and the supervisor agrees or disagrees with the system decision, resulting to a safe semi-autonomous interaction. The Wizard of Oz paradigm has been extensively used for RAT applications, where the robot executes the behaviors decided by a human supervisor. Despite its effectiveness, a main limitation relates to the amount of expert workload and attention to ensure a safe robot behavior. Towards this end, recent approaches enable the robot to learn through human (expert) input and progressively act in an autonomous manner.

Robotic agents can be learning or non-learning agents, or they can switch between these levels of learning, depending on different parameters (i.e., uncertainty). Active Learning is a research area which studies when an agent should ask for human input (i.e., correct label) in order to improve system performance, e.g., robot grasping [40]. Interactive Machine Learning and Interactive Reinforcement Learning are two promising approaches to integrate such human expertise and feedback in the learning mechanism of an interactive system (Human-in-the-Loop). Following such interactive learning approaches, intelligent WoZ interfaces can enable an assistive robot to integrate expert knowledge and guidance and switch from tele-operation to a progressively autonomous mode, decreasing expert workload and effort.

For example, neural networks have been used to learn robot behavior from human expert input in a RAT session [37]. The presented system simulates a RAT
session, where a human supervisor monitors a robot-child and a robot-instructor during a card classification task, using a WoZ interface. The neural network is trained using human input as training labels. Their user study results indicate that learning agents can decrease expert workload, as they learn how to provide human-like decisions. The robot shifts from a tele-operated agent (WoZ) to a fully autonomous robot, demonstrating that progressive robot autonomy results in lower supervisor workload.

1.3.4 Personalization Dimensions

Personalization plays an integral role in designing an efficient Robot-Assisted Training system. Based on the famous Bloom’s 2 sigma problem [41], one-to-one tutoring presents better learning effects than group (conventional) tutoring. Parameters that affect efficiency include training material (e.g., exercise regimen) and teacher behavior (e.g., supportive, challenging, etc.). Such parameters can be adjusted in order to maximize tutoring/training effects for each individual. Depending on the system parameters defined by the other taxonomy categories, we need to define the personalization dimensions. Personalization dimensions refer to (a) the set of observations that the system perceives and considers in order to adjust its behavior and (b) the set of control parameters that are adjusted to achieve personalization, as we show in Table 1.3.

One of the research questions regarding personalization is how to use these observations in order to learn the appropriate control parameters. Interactive Reinforcement Learning (IRL) techniques have been used to facilitate robot learning from human-generated feedback. For example, a robot that learns behavior by using the emotional (and other social) signals of the user could facilitate real-time personalization in human-robot interaction. For example, the affective language tutor [31], presented in Table 1.2, uses a facial expression and feature software in order
### Personalization Dimensions

<table>
<thead>
<tr>
<th>control parameters</th>
<th>observed parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>task difficulty</td>
<td>task performance</td>
</tr>
<tr>
<td>task duration and speed</td>
<td>user attention and engagement</td>
</tr>
<tr>
<td>supportive robot behavior</td>
<td>affective-emotional state</td>
</tr>
<tr>
<td>force and haptic feedback</td>
<td>response and completion time</td>
</tr>
<tr>
<td>training material</td>
<td>user progress</td>
</tr>
<tr>
<td>robot proxemics</td>
<td>errors</td>
</tr>
</tbody>
</table>

Table 1.3. Personalization Dimensions. A list of control and observed parameters for personalization to estimate child’s affective state (engagement and valence). The system combines these estimated values into a reward signal and the system learns to adjust its behavior by selecting appropriate motivational strategies (using verbal and non-verbal actions), based on current child’s state (affect and performance). However, it is of high importance to ensure a user-friendly and unobtrusive sensory system, selecting the appropriate (maybe minimum) set of sensors, considering factors as usability, effectiveness and efficiency.

1.4 Thesis Motivation and Outline

In this chapter, we presented a taxonomy in Robot-Assisted Training, considering related taxonomies in HRI, as well as current research works in RAT-based systems. The purpose of this taxonomy is to highlight several research objectives related to Robot-Assisted Training systems. We presented a systematic literature review, aiming to delineate different aspects and trends to be taken into consideration when designing a RAT-based system. Future improvements, updates and additions are required to establish a well-defined taxonomy in such a wide subarea of HRI, focusing on the needs and dimensions for personalization.
Robot-Assisted Training systems usually operate in contextually rich environments that can provide the system with valuable information to achieve personalization. A research question that arises is how to identify the optimal (e.g., minimum) set of modalities and sensors to ensure an efficient and effective interaction. As we discussed in Section 1.3.2, different interaction types and member roles result to different types of human feedback that can be captured by different sensors/interfaces including, cameras, microphones, EEG sensors, GUIs, joysticks, and many others.

These different types of feedback can be integrated to the system’s personalization mechanism, using interactive machine learning methods, towards interactive personalization. Interactive Machine Learning can utilize human-generated feedback (i.e., facial expressions, emotion, GUI input, etc.) in order to facilitate personalization in the wild, covering also rare user cases (e.g., unobserved or outlier users). Research works investigate how informative user interfaces and interactive learning methods can increase user engagement while interacting with a learning control interface [42].

Personalization is a complex computational problem that requires the training agent to interactively assess, adapt, and leverage a model of the user’s ability and needs [43] and can benefit from research reviews in several areas, including but not limited to, Intelligent Tutoring Systems [44], Student Modeling [45], Affective Computing [46], Cyber-Physical Systems [47] and Machine Learning for Interactive Systems and Robots [48].

The rest of the Thesis is outlined as follows. Chapter 2 discusses the interaction management problem of interactive agents, as well as how Reinforcement Learning (RL) can be deployed to model and optimize interaction patterns in the context of Robot-Assisted Training. As a use case, we present an adaptive rehabilitation session manager that uses an RL agent to adjust the exercise parameters in order to increase user performance and compliance. In Chapter 3, we present an Interactive
Learning and Adaptation Framework for personalized Robot-Assisted Training, which combines Interactive Reinforcement Learning methods to integrate different types of human feedback to facilitate the personalization of RAT-based system. In Chapter 4, we present a Socially Assistive Robot for personalized cognitive training, illustrating our proposed framework with the Sequence Learning task, as well as an initial system prototype. Chapter 5 discusses how human-in-the-loop and interactive machine learning approaches can be used to integrate human expertise (trainers, therapists) through Graphical User Interfaces (GUI), presenting a user study on user skill assessment using GUls. In Chapter 6, we present our data-driven approach towards developing a personalized SAR-based system that adjusts the training session based on task performance and engagement through EEG signals, using Interactive Reinforcement Learning. The evaluation of the system prototype and the final user study is presented in Chapter 6. Finally, this Thesis concludes with Chapter 7, in which we summarize and discuss our findings in order to provide possible future directions for this research area.
2.1 Personalization as Interaction Management problem

Interactive learning systems and robots are entities that learn through interacting with their environment and often need to perceive, act and communicate in a complex dynamic and uncertain environment [48, 49]. Interactions with interactive learning agents may involve complex signals through different modalities, including human-generated data (e.g., speech, gestures, eye-gaze direction, body motion, EEG signals, etc), as well as context-related observations (e.g., human performance).

Figure 2.1. The interaction loop with an interactive learning agent.

In order to enable machines to interact with users in a natural manner, the system must be able to identify or recognize patterns in human behavior and performance through the input modalities (e.g., speech recognition, emotion detection, gestures, etc.) and communicate through its output modalities (e.g., speech and
gesture generation). However, there is also need for higher level analysis and learn what to do based on the perceived data, in order to maximize the efficiency of the interaction [50]. This problem is known as interaction management; the problem of deciding what to do, based on perceived information and acquired knowledge from previous experience, in order to optimize the interaction. Based on the interaction loop, as shown in Figure 2.1, the agent must sense its environment, through the input multimodal modules, plan the next action, and act using its output processing modules.

Following this loop, personalization can be described as the process where the agent perceives and senses specific observations, based on which it adjusts its control parameters to optimize the interaction with respect to a specific parameter (e.g., task completion, user satisfaction, etc.). In order to model and optimize such interaction patterns, learning and adapting is an essential feature of any intelligent interactive system. As any learning system, its performance needs to improve over time, as it interacts with its environment [51].

Considering the sequential nature of interaction, interaction management and personalization can be seen as sequential decision making problems, where the system needs to make decisions about its next output based on prior experience and knowledge in order to maximize system’s performance. Even if interactive systems can be programmed and hand-coded to ensure an efficient and predicted behavior, such an approach is not robust to handle uncertainty and does not have a proper solution in realistic stochastic scenarios, since once the plan is decided, it can hardly be modified even if the interaction goes wrong. Machine Learning methods can be used to solve this problem and optimize the interaction patterns. Reinforcement Learning is an appropriate machine learning approach that can handle sequential decision making problems for systems that interact with dynamic environments.
2.1.1 Preliminaries in Reinforcement Learning

Reinforcement Learning (RL) is learning *what to do* – how to map situations to actions – so as to maximize system’s performance expressed as a numerical cumulative reward signal [52]. It provides an appropriate framework for interaction modeling and optimization for sequential decision making problems formulated as Markov Decision Processes (MDP). RL methods have been successfully applied for interaction modeling and optimization in Adaptive Dialogue Systems [53], Intelligent Tutoring Systems [54], Human-Robot Collaboration [55] and Robot-Assisted Therapy [33, 56].

Interactions within the RL setup, as shown in Figure 2.2, are sequences of states $S$, actions $A$ and rewards $R$. More specifically, the RL agent perceives state $S_t$ from its environment, based on which it selects to execute action $A_t$. The action is executed and the environment returns a new state $S_{t+1}$ and a reward $R_{t+1}$, which evaluates the current transition. The agent selects actions based on its *policy* $\pi$ which is a mapping from states to actions and dictates which action to execute given the current state. The goal of the agent is to interact with its environment by selecting actions in a way that maximizes future rewards.

![Figure 2.2. The Reinforcement Learning setup.](image)
An RL agent can be formulated as a *Markov Decision Process* (MDP). An MDP is described by a tuple $< S, A, T, R, \gamma >$ where:

- $S$ is a finite set of states – state space
- $A$ is the finite set of available actions – action space
- $T$ is the transition model where $T(s, a, s')$ denotes the probability $P(s'\mid s, a)$
- $R(s, a, s')$ is a reward function which evaluates the transition $s, a \rightarrow s'$
- $\gamma$ is a discount factor, $\gamma \in [0, 1]$

The main idea in RL is not to maximize the intermediate rewards after each transition, but the *expected total return* at the end of an episode. There are different reward models that give the total return. The straightforward one is to sum up all immediate rewards. In that case, it is hard to impose bounds on the sum, so the *discounted average reward* is preferred. In such case, a discount factor $\gamma$ is used to express the total return as the discounted sum of all intermediate rewards, such that $G = \sum_{t=1}^{T} \gamma^{t-1}R_t$, where $T$ is the number of interaction steps until the end of the episode. The value of the discount factor represents the difference in importance between future rewards and present rewards.

Reinforcement Learning methods can be divided into *model-based* algorithms, which learn a model of the system dynamics and plan based on it, and *model-free* techniques, which rely only on experience without learning a model [57, 58]. Although model-free methods avoid the need to model system dynamics, they typically require policies with carefully designed, low-dimensional parameterization [59]. On the other hand, model-based methods require the ability to learn an accurate model of the dynamics, which can be very difficult for complex systems, especially when the algorithm imposes restrictions on the dynamics representation to make the policy search efficient and numerically stable [60].
Sutton and Barto developed a unified view of model-based and model-free methods [52]. They consider the former technique as \textit{planning} and the latter as \textit{learning}. The most well-known learning approach is Temporal Differences (TD) learning. It is a combination of Dynamic Programming (DP) and Monte Carlo (MC) methods. The main idea is that instead of collecting full trajectory examples from simulated experiences (MC) and then compute the state-action values, TD methods can learn directly from raw experience without a model of the environment’s dynamics, by updating estimates based on previously learned estimates (DP), without waiting for a final outcome (bootstrapping). The goal of such algorithms is to estimate the \textit{Q-values}, or state-action values, based on which an optimal policy \( \pi^* \) can be defined. A Q-value is an estimation of the expected total return \( G_t \) starting from state \( s \), taking action \( a \) and following policy \( \pi \), \( Q^\pi(s, a) = \mathbb{E}[G_t|s_t = s, a_t = a, \pi] \). The optimal action-value function obeys an important identity known as the Bellman equation [52], where \( Q^*(s, a) = \mathbb{E}[r + \gamma \cdot \max_{a'} Q^*(s', a')|s, a] \). Model-free algorithms directly estimate the optimal Q-values from which the optimal policy \( \pi^* \) may be derived by choosing action \( a \) with the highest Q-value in the current state, i.e., \( \pi^*(s) = \arg \max_{a \in A} Q^*(s, a) \). The main difference between Q-learning and SARSA is that Q-learning makes the update based on the best possible policy (off-policy), while SARSA considers the actual next state and follows the policy it is learning (on-policy), as we show in Figure 2.3.

![Figure 2.3. Model-free algorithms: Q-learning (off-policy) and SARSA (on-policy).](image-url)
2.1.2 Reinforcement Learning for Robot-Assisted Training

Reinforcement Learning approaches have been extensively investigated and applied to model the interaction management problem (i.e., dialogue manager) for Adaptive Spoken and Multimodal Dialogue Systems, including Bayesian Reinforcement Learning [61], Deep Reinforcement Learning [62] and co-adaptation approaches [63]. Even in the area of HRI, robot behavior can be successfully modeled as a dialogue manager system [64], following specific guidelines for developing and evaluating RL-based dialogue systems in a data-driven manner [53]. This section discusses how Reinforcement Learning can be used to model robot behavior in a Robot-Assisted Training system, reviewing recent works of such RL-based systems.

Formulating such problems as Markov Decision Processes is one of the most crucial objectives towards developing an efficient RL-based interactive system. In the context of a Robot-Assisted Training system, as we show in Figure 2.4, the environment may consist of the user performing a training task, as well as the set of sensors used to capture changes in environment (task progress, user state, etc.). Robot actions are supposed to adjust parameters that affect system’s behavior, e.g., task difficulty, robot behavior, context parameters, and others.

Figure 2.4. Robot-Assisted Training as a Reinforcement Learning problem.
It is of high importance to model the system behavior such that it serves the purposes of the system. Designing an effective MDP (state-action space, reward function, transitions) may require empirical and iterative evaluations with system prototypes, user studies for data collection and analysis, as well as offline experimentation with simulated users [53].

One of the challenges, considering the state-action space, is the trade-off between level of dimensionality and scalability. On the one hand, a state-action space needs to be detailed and representative, even low-dimensional, in order to fully describe the dynamics and the purpose of the system. On the other hand, learning in large state-action spaces may not be feasible for real-time deployment. In the context of a RAT-based system the state space could include information about user’s state (cognitive, affective), as well as task parameters (task difficulty, task progress). For example, in the domain of affective personalization [31], the robotic affective tutor keeps track of tablet interaction events and student’s affective state. More specifically, the state features are valence, engagement (as estimated by an integrated software) and a binary value representing interaction with the tablet in the previous 5 seconds. The authors argue that the designed state space includes the required information in order to monitor and improve student’s affective state during a tablet-based interaction. On the other hand, training systems which focus on maximizing user’s performance may require other state features, such as user performance/progress, or information regarding user engagement and/or fatigue [32]. In a robot-based physical rehabilitation system [33], the state includes information regarding task parameters, i.e., task difficulty and speed. The authors propose a discretized state space small enough such that it can be fully explored in a real-time interaction, which is a main limitation of RL-based real-time systems. Function approximation methods, hierarchical representations and state abstractions have been used to overcome such burdens
in HRI systems [65], as well as methods to learn HRI models from limited data [66]. Such approaches should be investigated in the context or real-time personalized and adaptive HRI systems, to address issues related to scalability and feasibility [49, 50].

Another challenge considers the selection of an appropriate reward function, based on which the system will learn to optimize its policy. The decision of a reward function which can properly evaluate the behavior of the robot is a major challenge, and refers to both the reward amount, as well as when the reward should be provided. A reward signal can either depend on a single parameter or it can be a combination of parameters, either as a weighted sum or as an array of rewards. For instance, Multiobjective Reinforcement Learning (MORL) is a generalization of standard RL which can deal with multiple feedback signals and multiple objectives [67]. Moreover, rewards can be either intermediate or provided at the end of the episode. For example, in the robot-based rehabilitation system [33], the authors propose a reward, based on the state features (game difficulty and speed), which represents how far is the current user is from the hardest game level. The reward signal is the inverse euclidean distance of the current state to the hardest state. Such reward will guide the agent towards maximizing user challenge, by providing the user with the highest difficulty level the user can succeed. The affective robot tutor [31] uses a weighted sum of valence and engagement, as estimated by an integrated software, to formulate the reward signal. In another robot-based rehabilitation system, a social robot evaluates its motivational strategies, using a reward signal estimated based on the number of completed exercises averaged over a given period of time. Their approach takes into consideration both possible user fatigue over time, as well as distractions caused by the adaptation procedure [25].

These different approaches indicate the significance of designing an RL agent which efficiently models the main dynamics of the environment and the purpose of the
system, considering scalability and feasibility in real-time setups. Different representation and learning approaches could be investigated in the context of real-time HRI. Considering the state-action space, partially observable MDP (POMDP), as well as deep learning architectures [62] can be investigated to design more informative, but scalable policies. Moreover, Inverse Reinforcement Learning; an RL variation whose purpose is to learn a reward function given observed, optimal behaviors [68], could be used to learn robot behavior through human expert policies (e.g., therapist decisions, demonstrations). The task of learning from expert is called apprenticeship learning [69]. Moreover, Interactive Reinforcement Learning methods can be used to utilize human-provided feedback and knowledge in order to facilitate learning and adaptation. In this research, we focus on this latter method focusing on discrete state-action spaces. We investigate how Interactive Reinforcement Learning methods can be used to exploit different types of human feedback towards the personalization of a real-time Robot-Assisted Training system.

2.2 An RL-based Adaptive Rehabilitation Session Manager

In this section, we present our preliminary work on building an adaptive interactive agent using Reinforcement Learning, with a use case in Adaptive Rehabilitation. More specifically, we present ReADAPT, an adaptive module for a tele-rehabilitation system that takes into consideration multisensing data to adjust the session difficulty for each user resulting to a personalized session. Multimodal data such as speech, facial expressions and body motion can be collected during the exercising and feed the system to decide on the exercise and session difficulty. We formulate the problem as a Markov Decision Process and apply a Reinforcement Learning algorithm to train and evaluate the system on simulated data, in order to evaluate our RL-based approach.
2.2.1 Background and Related Work

Physical exercising is an essential part of any rehabilitation plan. The subject must be committed to a daily exercising routine, as well as to a frequent contact with the therapist. Rehabilitation plans can be quite expensive and time-consuming. On the other hand, telerehabilitation systems can be really helpful and efficient for both subjects and therapists.

![Figure 2.5. A telerehabilitation setup. A virtual agent acts as an exercise coach. The patient performs the prescribed exercise and the system monitors user performance. Courtesy of Swedish Health Care, RoboBusiness Leadership Summit, 2012.](image)

Advances in telerehabilitation systems including virtual reality, wearables, virtual agents, haptic interfaces have been proposed to overcome these burdens [70, 71]. Such technologies are being utilized to transform the traditional physical exercising and give the opportunity to the subject to perform the prescribed exercises from their own environment, while giving essential feedback and data to the therapist to enhance their decision making [72]. An example of such a telerehabilitation system that uses avatars and sensors to monitor the subjects performance is shown in Figure 2.5.
Such systems collect and process data from various sensors to monitor the performance of the subject while exercising. Such a system must be dynamic and adaptive, utilizing meaningful data from the subject during the rehabilitation session [73, 74]. Subject preferences and performance must be monitored to maintain an appropriate exercise difficulty level. Due to the repetitive nature of most of the rehabilitation exercises, the subject may not be easily engaged over longer periods of time and more likely to quit. This is especially true if the difficulty level of the exercise is predefined and static.

Making the exercise regimen adaptive in real time is a computational challenge. ReADAPT utilizes multisensory data such as body motion data, facial pain expressions, and speech as well as session information such as time spent on each exercise and the performance on the current exercise difficulty level. All this information is used as feedback to the system to modify and adapt the exercise category and difficulty. The purpose of such a system is to ensure that subjects are consistently and correctly performing their rehabilitation exercises according to the specified, by the therapist, protocol at home under the (remote) supervision of a therapist, while monitoring the pain levels to ensure safety and compliance. ReADAPT provides the therapist with a general framework to define a specific exercise protocol and the various system parameters based on the subjects needs and personalized plan. It enhances the therapists decision making by suggesting the exercise difficulty level, in order to achieve a high performance and subject compliance to the exercising plan.

2.2.2 Approach and Methodology

We have proposed an architecture of a Dialogue System employed to facilitate a safe interaction with a user during a rehabilitation session [75]. Dialogue Systems (DS) are able to interact with their users in a natural manner, typically using spoken
natural language, as well as other modalities (facial expressions, body motion, etc.).

In Figure 2.6, we show the architecture of the proposed dialogue system.

![Figure 2.6. The proposed Dialogue System for Safe Rehabilitation.](image)

Regarding robust communication with the system, we employ state of the art Audio Visual Automatic Speech Recognition (AVASR) algorithms, paired with a Natural Language Understanding (NLU) component, responsible for making sure the subjects response is on-topic and satisfactory. Moreover the system uses Text-to-Speech (TTS) and Natural Language Generation (NLG) to support an efficient interaction.

In this work [76], we present ReADAPT, a prototype of the dialogue manager (DM) of the system. The goal of ReADAPT is twofold; to engage the subject to complete the whole session of the exercises while preventing him/her from injuries (*maintain safety*) and high pain levels (*maintain compliance*). In order to model and optimize the adaptation module, we formulate the problem as a Markov Decision Process and apply Reinforcement Learning.

In order to achieve an effective interaction and session, the session manager must learn the optimal policy $\pi$; which defies the mapping from states to action that
maximize the total accumulated reward at the end of each session. Each session consists of many interaction steps. Each interaction step is the proposed exercise and difficulty and the exercise execution. A session ends if the system reaches a final state. A final state on this system can be either a goal state, where the subject completes successfully the whole set of exercises, or a quit state, where the subject decides to quit. We follow the strong assumption that the subject is more likely to quit or be non-compliant under high levels of pain. The interaction protocol (as an MDP) is shown in Figure 2.7.

![Figure 2.7](image)

Figure 2.7. ReADAPT manager formulated as a Reinforcement Learning problem.

During each session, the manager perceives multisensing information (speech, facial expressions, body motion, and session information) to define the current state $s_t$. Based on its policy and the current state, the manager performs an action $a_t$ and observes the new state $s_{t+1}$ and the reward $r_{t+1}$ based on the environmental feedback. For example, the manager may prompt the subject to perform the same exercise at a higher difficulty level and receive feedback, i.e., multisensory subject data, that the subject does not perform the exercise correctly. The manager receives a numerical
Table 2.1. ReADAPT as a Markov Decision Process

<table>
<thead>
<tr>
<th>State Features</th>
<th>System Actions</th>
<th>User Actions</th>
<th>Reward Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exercise ID</td>
<td>Continue Same</td>
<td>Perform Exercise</td>
<td>Exercise Level</td>
</tr>
<tr>
<td>Exercise Level</td>
<td>Level Up</td>
<td>Provide Pain Self-Report</td>
<td>Exercise Duration</td>
</tr>
<tr>
<td>Exercise Correctness</td>
<td>Level Down</td>
<td>Quit Session</td>
<td></td>
</tr>
<tr>
<td>Exercise Duration</td>
<td>Next Exercise</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Facial Pain Detection</td>
<td>Ask for User Pain Self-Report</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Pain Self-Report</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Self-Report Age</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quit Signal</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

reward that describes how good the selected action was and updates its knowledge for this specific state-action pair, by updating the Q-value.

Our approach follows the following scenario: The subject must complete a set of three prescribed exercises. Each exercise has three difficulty levels (Normal, Medium, Easy). Each exercise starts at the Normal level. The system keeps track of the time spent on each exercise using time units, $t_1 - t_5$. The specific mapping from actual time (or repetitions) to time units shall be defined by a professional therapist. During the exercise execution, the system measures the exercise performance compared to the ground truth given by the therapist, by analyzing the body motion capture of the subject (Good, Medium, Bad). Also a pain detection module is used to decide if there is a facial pain expression or not (No, Yes). An important state feature is the user pain report. The system learns to prompt the subject, asking for comfort and
pain levels, aiming to get an explicit answer. Responses are translated into a numeric value (0-11), according to the Numeric Rating Scale (NRS-11), an 11-point scale for subject self-reporting of pain. Collecting such information during real interactions can be very useful for improving the systems pain detection accuracy. Moreover, we include the age of the user pain report (in time units), since recent pain reports are more significant than older.

The system considers the multisensing input in order to decide on the next action. The systems available actions can modify and adjust the session by changing either the exercise parameters (difficulty, time) or by asking the user to provide their self-pain report. The system takes such actions in order to:

1. Engage the subject to complete the whole set of exercises correctly; if the subject does not execute well one exercise the system learns to adjust the difficulty to help the user.
2. Prevent the subject from incurring an injury and/or high levels of pain; by utilizing the facial pain recognition and the user self-report.
3. Spend the appropriate amount of time on each exercise; the system keeps track of the time spent on each exercise, so if a lot of time is spent in one exercise, then the system prompts the user to move to the next exercise.

An important part of the problem formulation as a Markov Decision Process is the reward function definition. It is responsible for evaluating the state-action pairs. For this approach, we designed the reward function to be dependent on the state features that describe the exercise level and the duration of each exercise. In Table 2.1, we summarize the MDP definition.
2.2.3 Experimental Procedure and Discussion

A significant weakness of RL approaches is the lack of data needed in order to estimate a transition model or to implement a reliable and accurate user model. In order to make an initial evaluation of our system, we have manually defined probability models to express the user responses to the session modifications based on the system actions. More specifically, we have defined: (1) a Real Pain (RP) model, (2) a User Pain Report (UR) model, (3) an Exercise Correctness (EC) model (4) a Visual Pain Detection (VP) model and (5) a User Quit (Q) model, as we show in Figure 2.8.

![Table of Simulated User Models]

Figure 2.8. Simulated user models.
In order to simulate as accurately as possible the pain reaction of the subject while performing the exercises, we define a model for the subject/user real pain. However, the state variables that refer to the subjects pain level are the user pain report and the visual pain detection. These variables cannot be defined in a deterministic manner. A user that suffers from high level of pain may provide the system with different user pain reports due to underestimation or overestimation. Moreover, we need to take into consideration the error possibility of the visual pain detection module. In particular, the real pain model gives the probability $P(RP|EL, RP_{pr})$, where $RP \in [0, 1, 2, 3]$ is the variable indicating the real pain level, $EL \in [1, 2, 3]$ is the variable indicating the exercise difficulty level and $RP_{pr}$ indicates the real pain level during the previous exercise execution. We make the assumption that the level of pain depends on the exercise level and the pain level experienced previously. In other words, a more demanding exercise has a higher probability to show or lead to higher than normal levels of pain, if there was a high level pain (i.e. $RP_{pr}$) experienced previously, than if the exercise started without any previous high pain.

One of the possible actions of the manager is to ask the user for their own pain report. The pain report is translated into a numerical likert scale. As mentioned in the real pain model section above, the user pain report depends on the actual pain of the subject. Based on the real pain level, the model gives the probability $P(UR|RP)$, where $UR \in [0, 11]$ is the variable that indicates the user report in the likert scale and $RP$ is the real pain level defined by the real pain model. The significance of using this model is the consideration of the human factors affecting the computation. The subject can misestimate their actual pain level. The basic reason is that pain cannot be measured and classified into a specific numeric level deterministically, since it is subjective. Moreover, other factors, such as psychological or physiological condition, led us to define the $UR$ variable dependent on the real pain level.
An important state variable of MDP is the feedback of the pain detection module using facial features. In [28], a pain detection system is described, that uses shape and appearance facial features. The accuracy of their system reaches 90.2% using decision trees. Despite this high accuracy, it still has an error possibility. For this reason, we define a visual pain model that gives the probability $P(VP|RP)$, where $VP$ is a binary variable indicating the presence or absence of pain. Based on the high accuracy that the pain detection system achieves, we assume that if the real pain level is the minimum or maximum, the pain detection system will be accurate. The probabilistic model we define is suitable to handle the uncertainty of the pain detection system for the mid-levels of pain. We have to mention that such models are used in order to make an initial evaluation of our proposed approach. Through the collection of real user interaction data, the system can learn a more robust learning model for our systems user.

A valuable feedback for the session manager is the exercise execution correction by the subject-user. If the subject under-performs or executes an exercise in an improper way, differently than the one prescribed, the system must adjust the difficulty level in order to enable the subject to perform the exercises correctly. The subjects performance while executing an exercise depends on the exercise difficulty level and the pain level. There are studies that show a high association between bodily pain, pain catastrophizing and exercise performance [29]. Taking into consideration the association between exercise performance with pain level and difficulty level, we define the exercise correctness model which gives the probability $P(EC|EL, RP_p)$, where $EC \in [0, 1, 2, 3]$ is the variable indicating how well the subject executes the exercise and $EL \in [1, 2, 3]$ indicates the difficulty level of the current exercise.

One possible user action during the interaction with the system is Quit. The subject may quit for several reasons. Some of them could be non-compliance, fatigue,
even loss of interest. We propose to overcome these burdens by adjusting dynamically the exercise difficulty as mentioned in previous sections. We also make the assumption that the subject is more likely to quit the session, if they are on high pain levels. In order to express this dependence between quitting and pain, we define the user-quit model that depends on the subjects pain level. This model gives the probability $P(Q|RP)$, where $Q$ is a binary variable that indicates if the user wants to quit the exercise. During real interactions, such system will be able to collect data in order to define a more accurate and personalized user model that will be expressing the probability of quitting the session depending on various multisensing data describing physiological and psychological factors, such as current mood, performance and progress, facial expressions and speech data, which can be used for long-term adherence.

For this initial implementation of the proposed prototype simulation system, we applied the Dyna-Q algorithm, following the $\epsilon$-greedy policy with linearly decreasing exploration rate. For each action selection, the algorithm performs $M = 100$ offline simulation steps, using the model it build during the interaction. After $N = 30000$ iterations, we evaluate the algorithm results. What we are interested in is to minimize the number of quits and maximize the average discounted reward. In order to visualize the results, we plot the number of quits and the average discounted reward for each 1000 sessions in our GUI shown in Figure 2.9.

We observe that the number of quits decreases as the algorithm learns the optimal policy. Moreover, the average discounted reward increases as the algorithm learns. The results are promising and are evidence that the manager learns by experience which action is best for each state in order to keep the subject safe and compliant to the rehabilitation session, by adapting the session exercise and difficulty or by asking the subject to self-report.
As a future extension, Wizard of Oz experiments can be conducted. A Wizard of Oz experiment is a research experiment in which subjects interact with a computer system that subjects believe to be autonomous, but which is actually being operated or partially operated by an unseen human being. The Wizard of Oz technique enables unimplemented technology to be evaluated by using a human to simulate the response of a system. Data will be recorded during these interactions and will be then used as training data for the learning algorithms. After the algorithms have been trained, we will conduct a second round of experiments in order to evaluate our system with human users who are not subjects as well as with trained therapists to gain intuition from their perspective. In future work, we will collect data during real interactions to evaluate how encouragement or breaks can assist the subject in terms of physiological and psychological state. An important contribution of such a system could be the collection of the real interaction data that will be annotated and combined with
data from different modalities. Such data can be used for modeling the user pain-based reactions during exercising. Using exercise performance data can also lead to a valuable research resource an annotated multimodal human activity repository. Finally, this work is important in a psychological setting. Audiovisual descriptors [35] can be used to associate exercise performance with psychological conditions in order to evaluate how emotional and mental states can affect compliance to physical performance, especially in the workplace.

In real-world systems, where the state-action space is large and the environment is not stationary, but its dynamics are likely to change over time, learning an optimal policy for different users is challenging. First of all, an interactive agent designed to interact with different users should be able to adapt to environmental changes by efficiently modifying a learned policy to cope with different users, instead of learning from scratch for each user. Furthermore, even the same user may not be consistent over time, in terms of reactions and intentions. In addition, user abilities change over time, adapting their own behavior while interacting with a learning agent. Recent works investigate the aspect of co-adaptation in man-machine interaction systems, assuming that both agent and user adapt in a cooperative manner to achieve a common goal [63, 77].

Taking these points into consideration, a dynamic adaptation mechanism is required for an interactive agent that enables the system to continuously adapt to the current user, ensuring efficient interactions. In the next section, we show how Interactive Reinforcement Learning can be used to tackle the challenge of real-time adaptation, by integrating human-generated feedback and expert guidance in the adaptation process. We propose an Interactive Learning and Adaptation framework and we illustrate it with a use case in Robot Assisted Training.
CHAPTER 3

INTERACTIVE LEARNING AND ADAPTATION FRAMEWORK

3.1 Introduction

A main advantage of RL methods in modeling interactive learning systems (i.e., Adaptive Dialogue Systems) is that the stochastic variation in user responses and the dynamics of the environment can be depicted as transition probabilities between states and actions in a Markov Decision Process [53]. However, in real-world systems, where the state-action space is large and the environment is dynamic, learning an optimal policy for each different user is challenging.

An interactive agent designed to interact with different users should be able to adapt to environmental changes by efficiently modifying a learned policy to cope with different users, instead of learning from scratch for each user. Furthermore, even the same user may not be consistent over time, in terms of reactions and intentions. In addition, user abilities change over time, as users adapt their own behavior while interacting with a learning agent. Recent works investigate the aspect of co-adaptation in man-machine interaction systems, assuming that both agent and user adapt in a cooperative manner to achieve a common goal [77]. A key challenge of applying RL to interactive systems is ensuring a safe interaction while adapting the agent’s behavior to a specific user, especially in sensitive environments, where exploration-based learning is not desirable [37].

Taking these points into consideration, a dynamic adaptation mechanism is required for an interactive agent that enables the system to adapt to different users, ensuring efficient interactions. In this chapter, we present an interactive learning and
adaptation framework for Robot-Assisted Training, which utilizes Interactive Reinforcement Learning methods to facilitate the policy adaptation of the robot towards new users. We argue that interactive RL methods [78, 79] can be used to facilitate the adaptation of an agent to a new user, utilizing human-generated feedback and human expertise.

3.2 Personalized Robot-Assisted Training: Methodology and Approach

In this section, we present an adaptive training task example, where the user needs to perform a set of cognitive or physical tasks. The representation we present is applicable to various RAT applications, as rehabilitation exercises and cognitive tasks [76, 33, 56]. We follow a scenario where the user interacts with the robot during a training session. The user must complete a set of three predefined tasks. Each task has four difficulty levels (Easy, Medium, Normal, Hard). The robot keeps track of task duration and the user’s score. To measure user’s score, we follow a negative marking approach. At each interaction step, the user receives a positive score proportional to the task difficulty, upon a successful turn and the corresponding negative one for failure. The robot keeps track of these scores and sums them to compute the user’s total performance.

3.2.1 Robot Behavior Modeling

In order to model the robot’s behavior, we formulate the problem as an MDP. The state is a set of variables that represent the current task state: Task ID [1, 2, 3], Task Duration [0 − 6], Difficulty Level [1 − 4] and Score [−4 − 4]. The score represents success or failure on a given level. At each interaction step, the system performs an action which sets the difficulty level for the current task and the user performs the task. The user receives a positive score for success and a negative one for
failure, proportional to the difficulty level. Based on the outcome, the agent receives a reward related to user’s score. The agent needs to learn the optimal policy; the mapping from states to actions that maximizes the accumulated reward (or total return) during each interaction. Maximizing total return can be translated as providing the appropriate difficulty level that will maximize user’s performance.

In order to evaluate our robot behavior modeling, we defined four different user models that capture different user abilities. These user models depict the user skills under different game parameters (task difficulty and duration). Each user model is a rule-based model whose binary output indicates user’s success (or failure) for each task difficulty and duration parameters. In Fig. 3.2, we show the defined user models that capture different user skills. The agent must learn how to adjust the difficulty level and when to switch to the next task, for each user model, learning a user-specific policy (USP).
3.2.2 Learning Experiments

Our first step is to train the agent against the four different user models. For our learning experiments, we applied the Q-learning algorithm, following the $\epsilon$-greedy policy with linearly decreasing exploration rate. An episode is a complete game of three tasks. In Fig. 3.3, we show the learning results for each user model. We visualize agent-based and task-based metrics; we plot the total return collected during each episode, averaged over a number of episodes (1 epoch = 50 episodes). Since the agent must learn the best policy for each user so as to maximize their performance,
we observe the similarity of total return and performance curves. Another metric is the start state value, which provides an estimate of the expected total return the agent can obtain by following this policy. In the top right figure, we observe the different convergence points for each user model. Since start state value expresses the expected return, we observe that start state value and average return tend to approximate each other as training evolves. Another convergence evaluation metric is the Q-value updates of all action-state pairs per epoch, showing that the algorithm converges as it learns, decreasing the state value updates.

Figure 3.3. Learning experiments. Applying Q-learning for the different user models results to different user-specific policies (USP)...
3.2.3 Policy Transfer Experiments

Our next step is to evaluate these four USP policies to the different user models. We make two hypotheses: (1) a user specific policy is the optimal policy (for the corresponding model); the one that maximizes total return, thus user performance, and (2) applying a learned policy to a different user model may not be efficient but better than learning from scratch. We applied the four different USP to the four different user models, following an exploitation-only approach, since following an exploration strategy may not be safe for real-world HRI applications.

Figure 3.4. Policy Transfer experiments. In this experiment, we applied all learned USP to the different user models.
These policy transfer experiments validate our two hypotheses (Fig. 3.4). Each USP is the optimal policy for the corresponding model; it maximizes the total return. Moreover, applying a policy to a different user model may not be efficient but better than learning from scratch (dashed-line). We can observe three cases: (1) the initial policy adapts and converges to the USP, (2) the initially policy is improved but does not converge to the USP and (3) the learned policy remains unchanged. On the bottom right figure, we observe that all USPs adapt and converge to USP4. This happens because the agent interacts with different user models, receives negative rewards for specific state-action pairs, improving its policy for the corresponding pairs. However, on the top left figure, we observe that all policies remain unchanged. This happens because all policies result to positive rewards, since user-1 succeeds in all difficulty levels. However, only USP1 is the optimal policy for this user model. This indicates a need for a dynamical adaptation mechanism that enables the agent to efficiently refine its policy towards a new user.

3.3 Interactive Learning and Adaptation Framework

In this section, we present an interactive learning and adaptation framework that integrates Interactive Reinforcement Learning approaches to the adaptation mechanism. Interactive Reinforcement Learning (IRL) is a variation of RL that studies how a human can be included in the agent learning process. We discuss how Interactive RL approaches can be used for personalization in a Robot-Assisted Training scenario. We present our use case, along with an initial evaluation of our proposed framework.
3.3.1 Interactive Reinforcement Learning

While Reinforcement Learning is not traditionally designed for interactive supervisory input from a human teacher, several works in both robot and software agents have integrated interactive feedback in the learning process, letting human users build classifiers for classical machine learning problems [43], guide a system to solve sequential tasks [80] or allow a domain expert to intervene in the learning process by providing demonstration examples to a robot [81]. A general framework for integrating a human trainer into the Reinforcement Learning framework is known as *Interactive Reinforcement Learning (IRL)*, as we visualize in Figure 3.5.A.

![Interactive Reinforcement Learning approaches.](image)

Interactive Reinforcement Learning studies how a human user can facilitate agent learning (1) by providing rewards in response to past actions, (2) by interven-
ing with anticipatory guidance to guide the selection of future actions (or prevent the execution of current bad actions) or (3) by providing the system with task demonstrations based on which the system will derive a policy by approximating human demonstrations. There are two main features that define these different approaches: (1) what type of signal is communicated and (2) when this signal is integrated to the learning process. Table 3.3.1 shows a classification of the three learning approaches: feedback, guidance and demonstration, based on these attributes.

<table>
<thead>
<tr>
<th></th>
<th>what</th>
<th>when</th>
</tr>
</thead>
<tbody>
<tr>
<td>feedback</td>
<td>additional reward</td>
<td>after execution</td>
</tr>
<tr>
<td>guidance</td>
<td>proposed action</td>
<td>before execution</td>
</tr>
<tr>
<td>demonstration</td>
<td>interaction examples</td>
<td>before interaction</td>
</tr>
</tbody>
</table>

Table 3.1. Interactive Reinforcement Learning Approaches

Learning from Demonstration (Figure 3.5.B) uses human demonstration samples to approximate a policy based on which the agent will interact with the environment [82]. Learning from Guidance (Figure 3.5.C) allows human intervention to the selected action before execution, proposing (corrective) actions [83], while Learning from Feedback (Figure 3.5.D) treats the human feedback as a reinforcement signal after the executed action [78].

These techniques refer to interactive systems, where the human trainer is not the primary user, but a secondary user that supervises the agent learning. To our knowledge, IRL methods have not been investigated for the adaptation of an agent to a new environment (user). We argue that human-generated feedback (Learning from Feedback) and human expertise (Learning from Guidance) can be provided by both primary and secondary users towards facilitating a safe and adaptive interaction [84].
Figure 3.6. We extend the RL framework by adding two additional communication channels; feedback and guidance. Their integration to the adaptation module can enable the agent to continuously adapt towards the current user, ensuring a safe and personalized interaction.

Based on our proposed framework (Figure 3.6), an interactive agent is able to adapt a learned policy towards a new user by exploiting additional communication channels (feedback and guidance). Our framework supports the participation of a secondary user who supervises the interaction in its early steps, avoiding unsafe interactions. The supervisor can either physically or remotely supervise the interaction. A user interface can be used to provide the supervisor with useful information about the agent learning procedure to help them monitor the interaction and enhance their own decision making, before altering the agent’s policy. The goal of this framework is
to enable agents to learn as long as they interact with primary and secondary users, adapting and refining their policy dynamically.

User feedback can be considered as a personalization factor, as it is provided by the primary user implicitly (e.g., facial expressions, attention, engagement or fatigue levels, etc.) and can be used to evaluate the interaction, thus the agent policy. Task engagement can provide valuable information to a system designed to provide challenging levels of difficulty. When the selected difficulty level is lower than needed, then the user may not be engaged. Moreover, when the task is difficult enough, the user may be frustrated and disengaged [85]. Sensor technologies, including cameras, EMG, EEG, eye-tracking, galvanic skin response and others can be used to extract such human-generated information. This implicit feedback can dynamically modify a learned policy towards a new user.

On the other hand, guidance can be considered as a safety factor, by integrating human advice to the adaptation mechanism. The therapist, as a secondary user, can set their own therapeutic goals by altering the policy either before or during the interaction. Making the learning process transparent to the secondary user may result to more informative guidance [86]. Informative metrics as state uncertainty and importance can be utilized to assist the secondary user provide the system with valuable guidance, in the form of corrective or suggested actions. Additionally, Active Learning methods [87] can be used to learn, based on state information, when the therapist should intervene, minimizing the expert’s workload as the system learns.

3.3.2 Adaptation Experiments

In this section, we present our preliminary adaptation experiments, following the proposed framework. As we mentioned, we assume that user feedback (engagement level) relates to the difficulty level variance; if the robot selects the appropriate
Figure 3.7. Integrating feedback and guidance to facilitate policy adaptation.

difficulty the engagement should be high, thus the feedback value. For our simulation experiments, feedback is the normalized absolute difference between the selected and the appropriate difficulty, so as $\text{feedback} \in [-1,0]$. The feedback is used to directly modify the Q-values, following the Q-augmentation technique [78].

On the other hand, we use guidance in the form of corrective actions, following a semi-supervised autonomy approach [37] combined with teaching on a budget [88]. Based on this approach, the agent proposes an action based on its policy. The therapist can reject this action and select another, for a limited number of interventions ($M = 2$). For our experiments, the corrective actions are selected based on the cor-
responding USP with probability 0.8, to cover possible therapist errors. In Fig. 3.7, we show the results of our experiments, integrating feedback and guidance.

3.4 Conclusion and Discussion

We observe that for all cases, the integration of feedback and guidance improve the applied policy, resulting to its convergence to the corresponding USP (optimal policy), validating our hypothesis that interactive learning methods can be utilized for the policy transfer and adaptation. These preliminary results are promising, since the integration of interactive learning techniques facilitate the policy adaptation. However, there are some limitations on the presented framework and simulation experiments. Based on the defined user models, users are consistent over time and towards all tasks. This is likely to be violated in a real-world scenario. However, we argue that the proposed framework can be evolved and applied to real-world HRI applications, investigating further how interactive learning techniques can be integrated to the adaptation mechanism, considering user inconsistency over time and co-adaptation. Moreover, we will investigate how users (both primary and secondary) interact under this framework, developing appropriate training tasks and conducting user studies.
CHAPTER 4
SOCIALLY ASSISTIVE ROBOTICS FOR COGNITIVE TRAINING

4.1 Background and Related Work

Socially Assistive Robotics (SAR) is a research area that studies how robots can be deployed to assist users through social interaction, as users perform a cognitive or physical task [89]. The goal of such assistive robots is to build an effective interaction with the user, so as to enhance their performance during the training session. Such agents can be deployed to various tasks, such as cognitive and/or physical training [90, 91], language tutoring [31], rehabilitation exercises [92] and others. As technology advances and becomes more affordable, SAR systems can be considered as an effective tool for educational and training purposes. A key feature of SAR systems is their ability to provide personalized interaction to the user. Personalization is essential for effective training or tutoring since it can enhance the effectiveness of the session, maximizing user learning potentials. A motivation for building intelligent robotic tutors arises from the famous Bloom’s 2 sigma result [41], based on which, one-to-one tutoring presents better learning effects than group (conventional) tutoring.

An effective robot-based training system should be able to adjust the difficulty of the task in order to provide a training session that fits user’s abilities and skills, resulting in a "optimally challenging activity” [93]. One approach is through behavioral and physiological monitoring, i.e., affect detection. Emotion and flow theories have been extensively applied to HRI applications. Considering the flow theory [2], affective states as boredom, engagement and anxiety can be detected through EEG sensors and used to adapt task difficulty in order to keep users in the flow channel.
Figure 4.1. Flow Theory [2] and Yerkes-Dodson Law [3]. Relation between user performance and skills, task difficulty and affective states (engagement, anxiety, etc.).

[85], as we show in Figure 4.1. Socially Assistive Robotics have been developed to improve user performance through the use of physiological signals [91], considering the Yerkes-Dodson law which links human arousal and task performance [3]. From another perspective, recent works define interactive personalization for socially assistive robotics as "the process by which an intelligent agent adapts to the needs and preferences of an individual user through eliciting information directly from that user about their state" [94, 95]. Based on this definition, certain information about the human learner may only be observable through learner’s direct input, as explicit feedback (e.g., self-report). Recent works focus on combining both implicit and explicit probes from the user considering task engagement, including self-reports, facial expressions and task behavior, towards developing a personalized engagement detection system [96]. Taking all approaches into consideration, personalization is a complex computational problem that requires the training agent to interactively assess, adapt, and leverage a model of the user’s abilities, skills, preferences, affect, etc., utilizing different types of feedback [43].
4.1.1 Reinforcement Learning for Socially Assistive Robotics

Socially Assistive Robots can provide personalized assistance through social assistance, by adjusting verbal, non-verbal or mixed behaviors (supportive feedback, attention acquisition, affective behavior, etc.) towards establishing an efficient interaction with the user. In this work, we focus on the personalization procedure of a SAR system, formulating it as a Reinforcement Learning problem. As mentioned before, Reinforcement Learning is an appropriate paradigm for learning sequential decision making processes with the potential to develop adaptive robots that adjust their behavior based on human abilities and needs, through either implicit or explicit feedback.

In the context of Socially Assistive Robotics, RL approaches are used to enable the robot personalize its behavior (i.e., policy) towards different users. Depending on the application, RL is used to adjust different parameters that can influence the effectiveness of the interaction. For example, in a language learning scenario, a social robot has been deployed to achieve personalization through affective behavior [31]. The presented system uses a camera to capture and analyze facial expressions and affective features (gaze, smile, engagement, valence, etc), during a language tutoring application, in order to provide a personalized affective interaction through social verbal behavior (valence and engagement of spoken instructions). The system combines the estimated values (user engagement and affect) into a reward signal. The system learns to adjust its behavior by selecting appropriate motivational strategies (using verbal and non-verbal actions), based on current child’s state (affect and performance), in order to keep the child engaged and in a positive affective state. Another example demonstrates how RL-based SAR systems can be deployed as exercise trainers, to enable personalized physical rehabilitation through social behavior adaptation [25]. In this work, the authors investigate three different robot behavior parameters.
(i.e., interaction distance, speed, and vocal content), and their effect to the user, in order to achieve long-term personalization and maximize user performance. The authors proposed a policy gradient RL (PGRL) method to learn the combination of the behavior parameters which maximizes user compliance and performance.

In another work, a social robotic tutor was proposed to assist users in logic puzzle solving [97]. The robotic tutor learns a user model during the interaction which assesses whether the user is experiencing difficulties in the task. Based on this, the robot decides whether it will perform a supportive behavior or not. An RL-based personalization module learns which specific supportive behavior (tangible, esteem, emotional support) can maximize user performance. RL-based personalization approaches has been also proposed for adaptive storytelling through social signals [98]. More specifically, the authors proposed an RL approach to learn which robot personality parameter (extraversion level) matches user’s preferences and keeps them engaged. The proposed system estimates user engagement, through a multisensing framework, and adjusts the robot’s current extraversion level to maximize user engagement during the session. Their simulated experiments show promising results in a small but noisy state space.

Social robots have also been used to engage individuals in cognitively stimulating activities through task-related assistance, using verbal and non-verbal behaviors [99], where the robot acts as a motivator during a memory card game. A hierarchical RL approach is used to enable the robot learn when to deploy specific assistive behaviors (assistance, encouragement and celebration), and personalize the interaction based on perceived user states (activity performance and arousal). In a similar application, RL has been proposed to dynamically adapt robot’s assistive behavior expressed by different modalities (i.e., speech, gaze, facial expression and head gesture) in a memory card game [34]. The system, under the guidance of a human wizard,
decides if the user needs help and selects the appropriate combination of gestures to
grab user’s attention, guide the user through the task and maximize task progress.

These works support the effectiveness of Reinforcement Learning as a personal-
ization framework for SAR-based systems. A main limitation of such RL-based
systems is scalability; learning efficiency and convergence speed when the state-action
space is large and the environment dynamics (human behavior), even the environment
itself (new user) change. Another limitation in designing RL agents for interactive
systems, including the definition of a proper state-action space, is defining an appro-
ropriate reward function that serves the purpose of the system [37, 100]. Our research is
motivated by the challenges that arise when different types of users and feedback types
are considered for real-time personalization using Reinforcement Learning [101]. To
this end, we illustrate the proposed Interactive Learning and Adaptation Framework
with a cognitive training task, investigating how Interactive RL methods (Learning
from Feedback) can be used to integrate human-generated feedback through EEG
data (task engagement) and facilitate personalization.

4.1.2 Brain Computer Interfaces

In a learning (tutoring, training) environment, affective and cognitive states are
highly correlated to task engagement and learning effects [102]. Positive states, i.e.,
flow, curiosity, and task engagement, have a positive correlation to learning, in com-
parison with negative states, such as boredom and/or frustration [103]. Taking into
consideration such information is essential in designing an effective learning or train-
ing system which estimates and monitors task engagement to adjust their behavior
parameters and sustain compliance [104, 105]. However, quantifying task engagement
and attention is not trivial, since it depends and overlaps with several user states,
such as interest, sustained attention, immersion and (attentional and emotional) in-
volvement [106]. Recently, Brain-Computer Interfaces (BCI) have been used towards this purpose [107]. There is a growing trend towards using passive BCI systems, which implicitly monitor brain activity, to personalize interactive systems through EEG sensors [108].

Figure 4.2. Muse headband and electrode locations by 10-20 international standard.

In our work, we follow the approach of passive BCI to measure and utilize task engagement, using the Muse EEG headset\(^1\). Muse is a low-cost portable EEG headset, which has been used to detect brain states of concentration and relaxation [109], task enjoyment [110] and for pain detection through self-calibrating protocols and interactive machine learning [111], as well as student cognitive state detection [112]. Muse provides 4 channels of data coming from dry frontal EEG electrodes (Fp1, Fp2, TP9, TP10), according to the 10-20 international standard. The device provides access to raw EEG signals as well as to a set of power spectral density measurements extracted from the raw data. The frequency bands provided by the device are \(\delta\) (1-4 Hz), \(\theta\) (5-8 Hz), \(\alpha\) (9-13 Hz), \(\beta\) (12-30 Hz) and \(\gamma\) (30-50 Hz). Research in EEG

\(^{1}\) http://www.choosemuse.com/
analysis for task engagement has offered the following formula for calculating a signal $E$, based on $\alpha$, $\beta$ and $\theta$ waves, which correlates to task engagement: $\epsilon = \beta/(\alpha + \theta)$ [113].

This approach has been followed for intelligent interactive systems that monitor task engagement and adjust their behavior to keep users engaged. In an adaptive storytelling application [108], a social robot used behavioral techniques (vocal cues, gestures) to regain user attention during drops in engagement, as estimated by the aforementioned formula. In a similar manner, this engagement index has been used to evaluate task engagement during a video game play [114]. The engagement index was capable of differentiating high intensity game events (e.g., player death) from general game play.

In our work, we outline the developmental process towards a data-driven SAR system that monitors task engagement and performance during a cognitive training task (Sequence Learning). We present an Interactive Reinforcement Learning approach that utilizes task engagement through EEG data to facilitate personalization. The long-term goal of this research is to develop HRI systems that combine and utilize different types of real-time human-generated feedback (implicit or explicit) to dynamically adjust their behavior in a lifelong learning environment.

4.2 The Sequence Learning Task

In this section, we present our experimental testbed, a cognitive task related to working memory and sequencing. Sequencing is the ability to arrange language, thoughts, information, and actions in an effective order. It has been shown that many children with learning and attention issues have trouble with sequencing [115].
Influenced by the NIH Toolbox Cognition Battery Working Memory (WM) test \(^2\) and SAR-based approaches for cognitive training [90, 25], we present the Sequence Learning task; a cognitive task, related to working memory, that evaluates the ability of an individual to remember and repeat a spoken sequence of letters. For our experimental setup, we deploy the NAO\(^3\) robot as a socially assistive robot that monitors both behavioral (task performance) and physiological (EEG) data and instructs the user towards a personalized cognitive training session.

\[ D = [1, 2, 3, 4] \]

\(^2\)http://www.healthmeasures.net/explore-measurement-systems/nih-toolbox/intro-to-nih-toolbox/cognition

\(^3\)https://www.ald.softbankrobotics.com/en/cool-robots/nao
is proportional to sequence length $L = [3, 5, 7, 9]$. For each sequence, the user receives a score, defined as:

$$score = \begin{cases} D, & \text{if} \, success = 1 \\ -1, & \text{if} \, success = 0 \end{cases}$$  \hfill (4.1)

Based on this scoring approach, the user gets more points (reward) by succeeding in harder levels, while the negative score is the same for all levels, such that the system does not discourage users from playing in harder levels. The robot can also provide feedback after the user completes a sequence. We examine the influence of different feedback styles and their relationship with user’s engagement and performance [116], since different users may prefer different feedback strategies, including the absence of feedback. More specifically, when the robot provides feedback, it reports the current outcome (success or failure), by providing either encouraging or challenging feedback and continues with a sequence of the same difficulty (length). In the case of no feedback, the robot moves on with the next sequence (same or different difficulty) without reporting on the result. Preliminary results show that absence of feedback can positively affect task performance in certain difficulty levels [117]. Examples of encouraging and challenging feedback are shown in Table 4.1.

<table>
<thead>
<tr>
<th></th>
<th><strong>Encouraging Feedback</strong></th>
<th><strong>Challenging Feedback</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>success</td>
<td>”That was great! keep up the good work”</td>
<td>”OK, that was easy enough! Let’s see now…”</td>
</tr>
<tr>
<td>failure</td>
<td>”Oh, that was wrong! But that’s fine! don’t give up!”</td>
<td>”Hey! Are you there? Stay focused when I speak!”</td>
</tr>
</tbody>
</table>

Table 4.1. Examples of encouraging and challenging feedback
4.3 Data Collection and Analysis

4.3.1 Data Collection

For the data collection procedure, we recruited 69 CSE undergraduate and graduate students, who received extra credits in their class, after agreement with their instructors. Each user completed a predefined session of the sequence learning task, consisted of 25 turns (sequences). Each session was sampled uniformly from a set of predefined sessions, such that the difficulty levels were near-uniformly distributed across all users. Each session lasted for about 20 minutes, including a post-session user survey.

Before the session, the participant was provided with a verbal and written explanation of the task and the experimental procedure. After the proper placement of the Muse sensor, the NAO robot greeted the user and provided them with a sequence example to get them familiarized with the task and the buttons setup, and ensure that the user did not require any more clarifications. From a preliminary user study on this task [117], we found out that users prefer to be aware of the upcoming difficulty level, before the robot announces the sequence. Considering this, the robot announces the difficulty level to the user, before each sequence.

At the end of the session, each user completed a user survey, regarding task difficulty and their self-assessment on task engagement and performance on each level. During each session, for each turn, we recorded the task parameters (turn ID, sequence length, robot feedback), user’s performance (user response, reaction and completion time) as well as the EEG data as provided by the Muse sensor. The EEG signals were evaluated and filtered based on the Muse headband status indicator, resulting in a dataset of 50 users (47 males and 10 females) between the age of 17 and 45 (M = 23.32, SD = 5.88).
full dataset, including the subset used for the following analysis and experiments, is publicly available online for researchers\textsuperscript{4}. The data is collected and stored such that they can be explored for several research purposes and approaches, including robot behavior modeling, user modeling, recommendation systems, EEG analysis and others. Considering the EEG data, we separate them based on the task state; while the user listens to the sequence (listening phase) and while the user responds (response phase). For the purpose of this work, we use the EEG data recorded during the listening phase.

4.3.2 User Survey

At the end of the experiment, the participants completed a survey for each level of the task. More specifically, the survey was designed to elicit subjective information from the users about task difficulty, task performance, task engagement and the reason for disengagement, if applicable. The results are shown in Figure 4.4.

![Figure 4.4. Survey Results during data collection.](https://github.com/TsiakasK/sequence-learning-dataset.git)
Considering task difficulty, more than 95% of the users find Level 1 and 2 either easy or just right. In contrast, only 44% and 8% of users find Level 3 and 4 easy or just right, respectively. Regarding task performance, 84% of users reported that they did above average in Level 1. This value changes to 70% for Level 2, and to 34% for Level 3. In Level 4, only 6% of users reported that their performance was either average or above average. Level 2 has the highest percentage of engaged users, as 96% of users reported they were engaged during this level, and this percentage is the lowest for Level 4 with only 60% of users reporting engagement during this level. This percentage goes to 76% for Level 1 and 86% for Level 3. The majority of the disengaged users in Levels 1 and 2 reported that task level easiness was the reason for their disengagement, and disengaged users in Levels 3 and 4 reported task difficulty as the reason for their disengagement.

4.3.3 User Modeling and Clustering

We analyze the performance and engagement data, in order to model different user behaviors across different task parameters. Our first step on the data analysis is to perform user clustering and group participants based on their task performance and engagement across the different difficulty levels. For each user, we estimate the engagement signal \( E \) using the engagement index formula [113, 108]. More specifically, the relative band values for alpha, beta and theta frequencies were extracted and smoothed, applying an exponentially-weighted moving average (EMWA) filter:

\[
\tilde{s}(t) = \begin{cases} 
  s(t) = y(t) & t = 0 \\
  s(t) = \alpha \ast \tilde{s}(t-1) + (1 - \alpha) \ast s(t-1) & t > 0
\end{cases}
\] (4.2)

These smoothed values were then used to estimate the engagement signal \( E = \alpha / (\beta + \theta) \), segmented per round and annotated by task difficulty, robot feedback.
and current result. For each user, this signal was normalized to [0, 1] and the mean engagement values were estimated for each difficulty level. Based on these, each user can now be represented with an array $UM = [P_1, P_2, P_3, P_4, E_1, E_2, E_3, E_4]$, where $P_{level=i} = P(success|level = i)$ and $E_{level=i} = \bar{e}_i$, where $\bar{e}_i$ is the mean engagement value for level $i \in [1, 4]$. In Figure 4.5, we visualize two different users and how they can be described by their performance and engagement values.

![Figure 4.5. Task performance and engagement for different users.](image)

The first row shows the normalized engagement signal during the session. The second row shows the mean engagement value for each difficulty level, and the third one visualizes task performance as probabilities of success at each level. We observe that User B can perform better in the task, since there is a high probability of success in the hardest level (Level 4), while for User A this probability is small, with Level 3 being the most difficult level this user can probably succeed. We observe that both users show their maximum engagement values during these levels (Level 3 for User A and Level 4 for User B).
In order to gain further insight into the distribution of the participants, considering performance and engagement, we project the user model arrays $UM$ into a 2D visualization using multidimensional scaling (MDS), with each point corresponding to a single user. We then apply K-Means ($K = 3$) clustering to the resulting projection, grouping the users into three clusters. The selection of $K$ is such that each cluster has an efficient number of samples ($\approx 15$). Each cluster can be seen as a group of users that share similar user skills and behaviors. Based on this, we visualize the cluster means; the average probability of success and mean engagement value per level. The results are visualized in Figure 4.6.

Based on the visualization of user clustering we could note that users in cluster 2 show a high probability of success in Level 4, relatively to users in clusters 1 and 3. Moreover, this cluster shows an upward trend in task engagement as difficulty increases. On the other hand, users in cluster 1 seem not probable of succeeding in levels higher than Level 2. However, their maximum task engagement values appear
for Level 3 and Level 4. As an example, the resulting clustering labeled User A as a member of cluster 3 and user B as a member of cluster 2 (see Figure 4.5).

A main limitation of the proposed system, considering our data collection procedure, is the limited amount of interaction data in a single session [66]. Long sessions (more than 30 rounds) would be inappropriate due to the cognitive effort required during the specific task. Our main consideration during data collection was to ensure (a) a near-uniform difficulty level distribution across users, as well as (b) different difficulty level transitions. To this end, as mentioned before, we used a predefined set of sessions. Considering the data visualization and analysis, we followed a baseline approach following the assumption that task performance and engagement depend only on the current level, disregarding task performance and engagement history. However, we argue that this approach serves our purpose to follow a data-driven methodology to learn personalized training policies for different users, assuming that these clusters depict three possible different types of users in terms of performance and engagement.

4.4 System Architecture and Prototype

The goal of this research is to develop an interactive learning and adaptation framework which utilizes prior knowledge (previous interactions) and human-generated data for real-time personalization. Moreover, the proposed framework supports the participation of a human supervisor who can guide the interaction through an intelligent monitoring and control interface. The framework utilizes these different sources of information (human-generated feedback, supervisor guidance and prior data) in order to provide a personalized session, in an interactive RL setup, as we show in Figure 4.7.
Figure 4.7. System Architecture for Real-Time Personalization.

The system consists of four main components: (a) the task environment, (b) a collection of user models and their corresponding personalized policies, (c) the supervisor GUI, and (d) the Interactive RL agent. The task environment refers to the physical setup along with the set of sensors needed to collect appropriate interaction data. The collection of user models can be described as a mapping from user models to personalized policies, which can be learned and updated both offline and online. The GUI component allows a human supervisor to communicate with the system for both monitoring and control purposes. Finally, the Interactive RL component can integrate information from the other components in order to provide a personalized SAR-based cognitive training session. A demo can be shown here\(^5\).

\(^5\)https://www.youtube.com/watch?v=XyHlw9iy1gc
For example, the RL agent can learn, update and utilize a set of models that capture different aspects, namely a performance model $P$, a feedback model $F$ and a guidance model $G$. The performance model captures user abilities over different difficulty levels. Similarly, the feedback model captures task engagement over different difficulty levels. Considering the guidance model, the system can learn a human input model based on the GUI interaction with a secondary user. The system can use this information to update its policy in order to personalize the interaction, as we show in Figure 4.8.

```
1: procedure Interactive Learning and Adaptation
2:   Initialize/Load performance model $P$
3:   Initialize/Load feedback model $F$
4:   Initialize/Load guidance model $G$
5:   Initialize/Load policy $\pi$
6:   $s = start\_state$
7:   while not done do
8:     Observe state $s$
9:     Select action $a$ based on $\pi(s)$
10:    Execute action $\bar{a}$ based on $G(s, a, g, P, F)$
11:    Observe user response, next state $s'$, reward $r$, and feedback $f$
12:    Update $P$, $F$, $G$, $\pi$
13:    $s = s'$
14:  end while
15:  Update user model collection and their corresponding policies
16: end procedure
```

Figure 4.8. System Procedure for Personalization.

Based on this, the system initializes its policy and $P$, $G$, and $F$ models. The system can either load existing policies and models or initialize them arbitrarily. The system observes the current state $s$ and selects an action $a$, based on the policy. The system parameters $(s, a, P, F)$ can be presented to a human supervisor (GUI) who can select an alternate action $g$ for the robot to perform. Based on the guidance model,
the system selects an action $\bar{a}$ to execute. In the case of a fully autonomous system, the guidance model would not affect system’s decision and action $a$ would be executed. However, if intervention is allowed, the guidance model can either return the actual human input $g$ or another action based on a supervised guidance model which learns from human input. A guidance model can be also predefined by a human user as a set of rules - which actions to take in specific states. The system executes the action and observes user response, next state $s'$, reward $r$, and feedback $f$. The system uses this information to update the models, as well as the policy. The system iterates until termination. The stored information can be used to update the user model collection and their corresponding policies. We need to note the this is an overall description of our proposed architecture with a large variety of possible modeling and learning approaches. For example, adopting the TAMER framework [32], the system can build a guidance or feedback model in a supervised manner and use the model to update the policy, following a Dyna architecture [52]. The framework can also function simultaneously with RL updates, combining MDP rewards with the human reward model [78].

For the purposes of our experiments, we designed a system prototype following the proposed architecture. Based on this prototype, the system has three modes: user skill assessment, policy selection and training, as we show in Figure 4.9. Based on this prototype, the system starts with the user skill assessment mode, based on which it records task performance and engagement under different difficulty levels in order to build a representative user model of the current user within the initial steps of the interaction. By the end of the assessment mode, the system has an indicative user model $UM$ for the current user. The system uses this model to classify the user into one of the existing user models, loading the corresponding user-specific policy, following the assumption that similar user models result to similar user-specific
policies [101]. This policy is loaded as the personalized training policy. At each interaction step, the system performs an action based on this policy, which is adjusted based on user feedback and human guidance. Prior knowledge (user models and user-specific policies), user feedback (task engagement through EEG) and human expertise (GUI input) are integrated to facilitate the personalization process, in an Interactive Reinforcement Learning setup.

In the next chapters, we describe the development and evaluation of this prototype version of the system. More specifically, Chapter 5 discusses how Human-in-the-Loop approaches can be applied to Robot-Assisted Training system. We present a user study on how informative interfaces can be used to allow human experts control and guide the assessment phase of the proposed prototype system. In Chapter 6, we
show our data-driven approach to learn a set of user models and their personalized RL-based training policies. We show how task engagement can be used as personalization feedback and in Chapter 7, we present our evaluation study with real users, discussing current limitations and further improvements.
CHAPTER 5

HUMAN-IN-THE-LOOP AND ROBOT-ASSISTED TRAINING

5.1 Introduction

Robot-Assisted Training (RAT) systems have been successfully deployed to provide assistance during a training task, promoting an efficient interaction with the user. Personalization can improve the efficiency of the interaction and thus enhance the effects of the training session. Personalization can be achieved through user skill assessment in order to choose an appropriate robot behavior that matches user abilities and needs. Intelligent user interfaces (IUIs) are human-machine interfaces that aim to improve the efficiency, effectiveness, and naturalness of human-machine interaction [118]. Graphical User Interfaces have been used to enable human supervisors to control robots and guide the interaction in RAT-based systems [119]. This chapter focuses on how such interfaces can be used to enable human supervisor users (e.g., therapists) to assess user skills during a robot-based cognitive task. In this study, we investigate how different visualization features affect decision making and efficiency, towards the design of an intelligent and informative interface.

5.2 User Skill Assessment using Informative Interfaces

User skill assessment is essential in order to achieve a personalized training session. However, assessing the skills of a new individual is not straightforward, since such agents often require a huge amount of interaction data in order to build a representative model of user skills. Human experts have the ability to assess users in an intuitive way, by identifying the specific skills that need to be assessed. SAR-based
systems can support the participation of a secondary user (e.g., supervisor, teacher, therapist), who can monitor and support the interaction between the assistive robot and the primary user, through graphical user interfaces [35].

We investigate how graphical user interfaces can be exploited to enhance the decision making of secondary users, considering user skill assessment. It has been shown that informative interfaces, i.e., interfaces that visualize task-related information, can increase user involvement and thereby improve systems performance [120]. On the other hand, human input can be leveraged and allow the system to learn from human guidance and act in a progressively autonomous manner, decreasing expert’s workload [37]. Our work moves towards the design of an intelligent and informative interface which allows a human supervisor to monitor and control a SAR-based cognitive training session. The purpose is twofold; (1) to investigate which interface features (e.g., visualization, transparency, etc.) enhance human decision making and improve system performance and (2) to leverage human input (GUI) and enable the robot learn from human input models. In this study, we investigate how visualization affects decision making and efficiency, in a user skill assessment task.

5.2.1 Related Work

In the context of RAT systems, graphical user interfaces can be used for both monitoring and control purposes. An informative interface can visualize essential information, i.e., task progress, user performance, task difficulty, and others. This information can be very useful as it can enhance human decision making. Such interfaces can be also used as Wizard-of-Oz interfaces to control robot’s behavior during the interaction. The Wizard-of-Oz robot control methodology provides an appropriate approach for human supervisors to control and guide the interaction between the robot and the primary user. This methodology can be utilized towards
designing autonomous HRI systems [121]. Recent work suggests that such wizarded robots can be learning agents and learn online from human input [37]. The authors presented a user study where participants were asked to control a robot during a simulated RAT session. Their findings support that learning interfaces can reduce human workload and improve system performance over time and can facilitate robot personalization.

Personalization is essential in RAT systems, since it can increase user compliance and enhance training effects. In a human-robot tutoring scenario, a robot tutor has been proposed to assist users in logic puzzle solving (nanogram) [122]. The robot tutor learns a user skill assessment model, using a Bayesian Network, and uses it to select the most appropriate next lesson that suits individual’s skills. Another approach for the same task is a Reinforcement Learning (RL) approach [97]. In this approach, the robotic tutor learns a user model during the interaction which assesses whether the user is experiencing difficulties in the task. Based on this, the robot decides whether it will perform a supportive behavior or not. An RL-based personalization module learns which specific supportive behavior (tangible, esteem, emotional support) can maximize user performance. In another application area, dynamic user modeling has been proposed to assess the level of expertise in a dialogue-based instruction scenario in order to adjust the system’s referring expression [123]. The dialogue system uses a dynamic user model to update its knowledge about user’s expertise in a given topic.

These works indicate the importance of learning a representative user model that assesses user skills to enhance user’s performance, in the context of a training task. Moreover, human guidance can be integrated to the system through graphical user interfaces to control robot behavior, while the robot can learn from human input. Towards this direction, we present our approach on how informative interfaces can be used as both monitoring and control interfaces during a robot-based cognitive task.
5.2.2 Methodology and Approach

Our goal is to develop an intelligent and informative interface that allows a human supervisor to monitor and guide the assessment phase of the system, as the robot interacts with a new user. In order to constrain our experimental setup, a user simulation model was used to stand in for a real human player. To this end, we use the clustered data to build user simulation models.

Based on the clustering (See Section 4), we built three user simulation models that capture different performance skills. For each cluster, we estimate the performance probabilities on each level. In order to deal with unobserved states, we deployed a neural network with softmax output, as a regressor, which estimates success probability for all levels and previous results. These different models can capture different user abilities considering task performance in different difficulty levels.

In the user study, 30 participants (undergraduate/graduate students) were asked to monitor and guide the NAO robot using an interface during the assessment phase of the sequence learning task. The participants are 24 males and 6 females with an age range between 20 and 34 years old. Most of the participants had no prior interaction experience with a SAR; only 6 participants have reported such an experience. In order to provide the participants with a realistic environment, the study administrators acted as the primary users. Participants were under the impression that the primary users actually performed the task, while they actually interacted with a user simulation model.

During the task, the participant selects a difficulty level using the GUI, the "fake" user presses the buttons (the participant can not observe which buttons), and the GUI visualizes the outcome (success, failure), based on the underlying user model. The goal is to assist participants to accurately estimate the behavior of the underlying user model, as we describe below. Each participant controls three short
assessment sessions (9 turns / session), each one with a different interface, following the protocol shown in Figure 5.1. At the beginning of the experiment, the participant was provided with a detailed description of the study and the goal of the assessment session. After the introduction, the participant performed three assessment sessions, each one under a different condition. There are three conditions: (1) control-only, (2) history-based monitoring and (3) model-based monitoring, one for each interface, as we show in Figure 5.2. The order of the conditions is predefined for each experiment, to ensure a uniform sampling of all possible sequences of conditions. To remove any bias effect, for each session a different "fake" user (underlying user model) performed the task.

For the control-only condition, the interface includes only the buttons that control the next difficulty level. The outcome (success/failure) is provided in written at the end of each turn. The history-based monitoring interface provides a visualization of the task history, considering previous outcomes and levels played. More specifically, the interface visualizes a plot where the x-axis is the number of turns and the y-axis is the score \( s \in [-4, 4] \), defined as \( s = \text{outcome} \times \text{level} \), where \( \text{outcome} = [-1, 1] \) and \( \text{level} = [1, 2, 3, 4] \). The model-based interface visualizes an estimation of the performance model \( P(\text{success}|\text{level}) \), where the x-axis represents the difficulty level and the y-axis the probability (frequency) of success at each level, updated after each turn. At each interaction step, the system uses human input (selected level) and
Figure 5.2. The participant interacts with three different interfaces during a simulated robot assisted training assessment session. The (a) control-only interface does not have any monitoring features. The (b) history-based GUI provides a history of task performance over past levels, while (c) the model-based interface provides a visualization of the performance model as a set of success probabilities at each level.

the previous result to estimate \( P(\text{success}|\text{level}, \text{previous\_result}) \), based on which it returns success or failure. While the "fake" user pretends to repeat the sequence by pushing the buttons, the visualized outcome is estimated based on the model. During the interaction, we record turn number, reaction time, human input (selected level), and model outcome. After each assessment session, the participant is asked to complete a session survey (Figure 5.3) that reports the session ID, an estimation of the user success rate at each level, if it was easy to judge the user’s performance
using the recent GUI and if the number of turns was enough. The session ID was provided to the participants without an explanation, and it is basically an indication of the condition and the user model, which were assigned to users in different orders to avoid the order effect. In the last session survey, the participant is also asked about their most enjoyed interface, which interface helped them the most to assess the users’ performance, their overall evaluation of the experiment and if they had any comments.

Figure 5.3. The participant completes a session survey after each assessment session.
For our evaluation, we consider and analyze both the survey results and the data recorded while participants using the three interfaces. Based on the survey results, the majority of the participants reported that the history-based monitoring was the most enjoyable and effective interface, followed by the model-based monitoring and the control-only monitoring, respectively (see Figure 5.4). Out of the 30 participants, 29 participants reported that it was *Easy* or *OK* to judge the user’s performance using the history-based interface, compared to 21 and 28 participants on the usage of the control-only and model-based interfaces, respectively. Also, more than half of the participants agreed that the 9-turns were enough to judge user’s performance using the different interfaces, with one of the participants commenting that the number of turns should allow them to assign an equal number of difficulty levels (i.e. the number of turns should be based on multiples of 4).

Figure 5.4. Participants' feedback on the enjoyability and the effectiveness of the three interfaces.
When comparing the actual user model outcomes and the (pre-calculated) user model estimates with the performance reported by the participants in the surveys, we found interesting results. Figure 5.5 shows that the mean squared error (MSE) of the performance evaluation between the survey and user models is large, which is expected as these pre-calculated user models do not necessarily fully represent the actual performance of the user models. The actual user models performance depends on both the pre-calculated performance and the difficulty levels selected by the participants. For instance, if the participant selects level 4 once and the user model reports *Failure*, then the actual performance is 0% even though the pre-calculated performance is 5%. The MSE between the survey data and actual data is much less, and it shows that the quality of participants evaluation is ranked the highest using the model-based GUI, followed by the history-based GUI and the control-only GUI, respectively.

![Mean Squared Error per GUI](image)

Figure 5.5. The mean squared errors between the survey data and both the actual data and the pre-calculated user models.
5.3 Discussion and Future Work

In this chapter, we presented a personalization framework for a SAR-based cognitive training task, focusing on how supervisor users can guide the assessment phase using an informative interface. For our experimental protocol, we developed a set of simulated user models that capture task performance over different difficulty levels of the sequence learning task. For our user study, participants were asked to use three different interfaces in order to guide the assessment phase and to use surveys to report on how the (simulated) users performed.

Our results indicate that the participants preferred the interfaces in the following order: history-based GUI, model-based GUI and control-only GUI, respectively. However, their evaluation quality is the higher using model-based GUI, history-based GUI and control-only GUI, respectively. With that in mind, the control-only GUI was the least preferred monitoring interface and resulted in worst evaluation quality. Based on the results, we can argue that both user’s history and a model-based visualization are important features for our proposed interface.

The long-term goal of this research is to develop an intelligent and informative interface to: (1) provide human supervisors with an intuitive and efficient visualization of user skills to enhance their decision making and (2) leverage human input to enable the robot to dynamically learn human-like policies and act autonomously (interactive learning). Considering visualization, we investigate different underlying user modeling approaches in order to provide the user with a more informative and intuitive visualization [122]. Since transparency is essential in building an effective interaction between the user and the interface, we investigate additional metrics, such as model uncertainty, that can provide a better understanding to the supervisor [120, 86].
In order to enable the robot leverage human input, we will follow our proposed framework for learning from guidance [101], where human input can be used to modify an RL-based policy and enable the robot act in a progressively autonomous manner. Additionally, Active Learning methods [87] can be used to learn, based on state information, when the therapist should intervene, minimizing the expert’s workload as the system learns. As part of preliminary work and our ongoing experiments [124], we have developed a prototype GUI for monitoring and control, as we show in Figure 5.6, which enables an expert user to monitor task performance and engagement during the interaction, visualize robot’s policy by highlighting next intended action and intervene by pressing the appropriate button in a real-time setup.

![Figure 5.6. A GUI prototype for Intelligent Monitoring and Control.](image)
CHAPTER 6

TASK ENGAGEMENT AS PERSONALIZATION FEEDBACK

6.1 Using Task Engagement as Personalization Feedback

As already mentioned, Learning from Feedback treats human input as a reinforcement signal after the executed action. Several works have considered the use of feedback to facilitate the learning procedure. In [78], they proposed the TAMER framework that includes a supervised learner for building a human reward model during the interaction, which enables humans to shape agents during their learning. In [79], they present a learning framework that integrates human feedback, in the form of facial expressions, as an additional reinforcement signal, to guide the agent during learning. In [125], the authors propose a method for personalized information filtering for learning user preferences, by capturing and transforming implicit feedback from the user to a reinforcement signal. These approaches support that IRL methods can facilitate real-time personalization from human-generated feedback.

There are two main approaches based on how feedback is integrated to the RL mechanism: reward shaping and Q-augmentation [78]. Reward shaping uses the feedback as an additional reward component added to the environmental reward \( R'(s,a) = R(s,a) + \beta \times F(s,a) \), while in Q-augmentation, feedback is used to directly adjust the policy, by modifying the Q-values \( Q'(s,a) = Q(s,a) + \beta \times F(s,a) \). A specific Q-value is an estimate of the long-term expected discounted reward for taking action \( a \) in state \( s \). In both techniques, \( F(s,a) \) is the feedback model and \( \beta \) is the combination parameter. In this work, we investigate how both techniques can be used to integrate task engagement into the learning process, and facilitate
learning, towards a personalized training session. Our approach for utilizing feedback for learning personalized policies is shown in Figure 6.1

![Image](image.png)

Figure 6.1. The Sequence Learning task as an RL problem.

The state space includes information about task difficulty and robot feedback, as well as previous result (previous level and outcome). More specifically, the state features are: Sequence Length – \( SL = [3, 5, 7, 9] \), Robot Feedback – \( RF = \{0 : \text{None}, 1 : \text{Encouraging}, 2 : \text{Challenging}\} \) and Previous Result – \( PR = [-4, 4] \). Based on the current state, the robot selects one of the available system actions \( A = [L1, L2, L3, L4, RF1, RF2] \), and the system perceives the next state, receiving a reward based on task performance and task engagement, as we describe in our experimental procedure. The transition model describes the dynamics of the task and captures how user performance varies from state to state. We need to note that
the state space is designed to be stochastic; each state might lead to a successful turn (positive score) with some probability, in order to capture different user abilities. Table 6.1 shows the state-action and reward components.

<table>
<thead>
<tr>
<th>State Features</th>
<th>System Actions</th>
<th>Reward</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence Length (SL)</td>
<td>Level 1 (L1)</td>
<td>Current Score</td>
</tr>
<tr>
<td>Robot Feedback (RF)</td>
<td>Level 2 (L2)</td>
<td><em>Task Engagement</em></td>
</tr>
<tr>
<td>Previous Result (PR)</td>
<td>Level 3 (L3)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Level 4 (L4)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Encouraging Feedback (RF1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Challenging Feedback (RF2)</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.1. The defined MDP of the problem

6.2 Learning Personalized Training Policies

As mentioned before, we follow a data-driven approach to get insights towards the development of our proposed SAR system. Data-driven methods are being used as methodologies for system development and evaluation of complex interactive systems, e.g., adaptive multimodal Dialogue Systems [53], including user simulation for offline RL experimentation, training, and evaluation.

![Figure 6.2. Reinforcement Learning setup using simulated users.](image)
In this section, we present our approach for user simulation and the RL experiments. Using the collected data and their analysis, we build simulation models considering task performance and engagement. Building simulated users allows for offline RL experimentation, as we show in Figure 6.2, before the system deployment with real users, towards creating a database of user models and their corresponding personalized policies.

6.2.1 User Simulation

For each user cluster $k$, we define the performance model and the engagement model. The performance model is defined as $UP_k = P(success|state)$, which estimates how likely a user is to succeed at a given state. To learn these success probabilities from our collected data, we employ a regression model on the observed data [126]. More specifically, for each cluster $k$, we apply Maximum Likelihood Estimation (MLE), to learn the probabilities $P(success|state) = \frac{N(success, state)}{N(state)}$ on our observed data. In order to deal with unobserved states, we deployed a neural network with softmax output, as a regression model which estimates success probability for all possible states. The input for the performance model is the state features: current level, robot feedback, and previous scores, and the output is the probability of success at this state.

In a similar way, the feedback model is defined as $\Phi_k(state, success)$ which estimates task engagement for each state and outcome, for each cluster $k$. The data extraction resulted to a set of number examples, respectively for each cluster. In order to train the model based on the clustered data, we estimated the mean engagement value per state and outcome. Support Vector Regression was used as a regression model to estimate the engagement value $\phi \in [0,1]$ for each state and outcome. As an
evaluation metric we used the training error. The root mean squared error (RMSE) was 0.08, 0.08, and 0.11, for each cluster respectively.

As an ongoing work, we are investigating further methods for EEG analysis [127] and user simulation, such as Input-Output Hidden Markov Models [128] and Dynamic Bayesian Networks [129] to also encode temporal information, which is out of the scope of this current work. In the next section, we describe the Reinforcement Learning experiments, following these baseline user simulation models.

6.2.2 Reinforcement Learning Experiments

The scope of the following reinforcement learning experiments is to investigate if integrating task engagement in the RL mechanism improves learning results, using the defined user models. The outcome of these learning experiments will be a set of user models (UM) and their corresponding user-specific policies (USP). To learn these policies, we apply Q-learning with learning rate $\alpha = 0.05$ and discount factor $\gamma = 0.95$ with softmax exploration strategy with state-visit based decreasing temperature parameter $\tau$. The small learning rate avoids instability in learning under noisy observations. The large discount factor enhances maximizing long-term rewards. The motivation for these experiments is to identify an appropriate method to utilize both task performance and engagement to learn personalized RL-based policies for real-time interaction.

For each cluster, we use the corresponding simulation models to learn the RL policies, as we show in Figure 6.2. Our first approach is to use a performance-based reward, where $r(s,a) = \text{score}$ (Eq. 4.1). In order to integrate task engagement, for reward shaping, we used $r'(s,a) = r(s,a) + \beta_1 * \Phi(s,a)$ and for Q-augmentation $Q'(s,a) = Q(s,a) + \beta_2 * \Phi(s,a)$. The selection of such parameter depends on the range of values (rewards and Q-values), as well as on patterns in performance and
engagement. After an empirical evaluation with different $\beta$ values, we selected $\beta_1 = 7.5$ and $\beta_2 = 0.8$. We present the learning results in Figure 6.4, visualizing task performance and task engagement for each user model, as the algorithm learns with each method.

![Graphs showing task performance and engagement for different user models](image)

Figure 6.3. Q-Learning results for the different user models. We visualize task performance and task engagement for each user model. Task engagement as personalization feedback can facilitate learning.

We observe that integration of task engagement through both feedback techniques increases both task performance and engagement, as the algorithm converges to the optimal policy. Generally, both techniques outperform the performance-based approach for the selected combination parameters. In order to get an insight of the learned policies, we visualize $p(\text{action}|\text{state})$ for a specific state, in order to compare...
the decisions across users and methods. The selected state is $S_0 = (5, 1, -2)$, based on which the current difficulty level is Level 2 ($L = 5$), and the robot has provided encouraging feedback ($RF = 1$) after the user failed in the same level on the previous turn ($PR = -2$).

$$\pi(S_0) = P(action|S_0)$$

Figure 6.4. Visualization of the learned policies for a given state: $\pi(state) = P(action|S_0)$. The x-axis shows the possible actions and y-axis shows the probability for each action. Each row corresponds to a user model and each column corresponds to the learning method.

Essentially, the figure visualizes the policies as the probabilities of selecting one of the actions $A = [L1, L2, L3, L4, RF1, RF2]$ on the given state. Each row corresponds to a user model and each column corresponds to the learning method. Considering a specific method, we can compare the USPs of different users. For example, considering reward-shaping, policy for *User Model 1* chooses action $L2$ with a high
probability, while for *User Model 2* the policy chooses almost uniformly between *L2* and *L3* and for *User Model 3* the policy is deterministic, choosing *RF2*. Considering a specific user model, we can observe differences in policies across methods. For example, for *User Model 2*, we observe that the performance-based approach results to a near-uniform policy, which does not provide enough information. On the contrary, both feedback methods result to a more informative policy, giving high probabilities only on two actions (*L2, L3* for reward shaping and *L3, RF1* for Q-augmentation). Both feedback methods propose higher levels to the user, which can increase task performance.

6.3 Discussion and Future Work

In this work, we presented the developmental process of a data-driven SAR system for personalized Robot-Assisted Training. The process includes data collection and analysis, the user simulation and RL experiments to integrate task engagement to the RL agent. This approach allows for an extensive analysis and experimentation, towards the development of a real-time personalized SAR system. The presented analysis and experiments indicate that users with different skills show different patterns in their task engagement for different difficulty levels. We argue that task engagement is an essential information that can be utilized for real-time adaptive SAR systems. Future work includes further experimentation in user performance simulation and EEG modeling, capturing patterns in task engagement under different user skills and difficulty levels, including temporal aspects (Input-Output HMM) [128]. More sophisticated and accurate user modeling will enable us to learn representative combination parameters and shaping functions. Moreover, a parametric mapping from user models to policies could enable the system handle rare user cases and outliers. User studies will be conducted to evaluate and refine the proposed system.
CHAPTER 7
USER STUDY AND FRAMEWORK EVALUATION

This chapter serves as an evaluation study of the proposed prototype system for SAR-based personalized cognitive training. The prototype system is designed and developed based on the outcomes from Chapter 5 (Graphical Interfaces) and Chapter 6 (set of models and personalized policies). The goal of the following study is to evaluate the proposed personalization approach in terms of task performance and user experience and gain further insights towards refinement and improvement.

7.1 Study Protocol

For the user study, we recruited 10 undergraduate/graduate students to perform two sessions of the task. The participants were 6 males and 4 females with an age range between 19 and 38 years old. Each session included 30 rounds: 10 rounds for assessment and 20 for training. We follow a within-study design approach; the study protocol and the two conditions are shown in Figure 7.1.

![Figure 7.1. Study Protocol for System Evaluation.](image-url)
The two conditions are (a) Random Training Policy and (b) RL-based Training Policy, which differ in the training mode. For both conditions, the assessment mode is wizarded; a human supervisor monitors and guides the assessment phase of the system through an interface, following the protocol described in Section 5. Since different supervisors may follow different assessment strategies, a single individual (study administrator) was in charge of all assessment phases.

At the end of the assessment phase, the system computes an indicative user model \( UM = [P_1, P_2, P_3, P_4, E_1, E_2, E_3, E_4] \), which provides an estimation of task performance \( P \) and engagement \( E \) for each level. For the random condition, the system ignores this model and follows a random (uniform) policy during the training mode. For the RL-based condition, the system uses the user model to classify the user into one of the three clusters, using KNN classification. Then, the IRL agent loads the corresponding personalized policy (user-specific policy) for the training mode. During the training mode, the RL agent updates online its policy using SARSA with reward shaping, as shown in Chapter 6. The parameters were empirically selected (exploration rate \( \epsilon = 0.5 \), learning rate \( \alpha = 0.5 \) and combination parameter \( \beta = 0.8 \)).

Each user performed both sessions in random order to eliminate the order effect. Due to the high cognitive effort required by the participants, the sessions were non-consecutive; a minimum break of 20 minutes was added between the two sessions. At the beginning of the first session, each participant was provided with a description of the experiment. The participants were told that they would interact with two different robot trainers at each session, without providing more information about their differences. After the proper placement of the Muse sensor, the robot introduced itself using a different name for each condition (”Sam” for the RL-based and ”Robert” for the random condition). After each session, participants completed a 5-point likert scale survey, as we show in Figure 7.3, to rate the degree to which they agree or
disagree with a set of statements, considering different aspects of the SAR-based interaction, e.g., self-report on performance and engagement, as well as evaluation of the provided difficulty levels and feedback. Moreover, participants rated the system in terms of 'intelligence', as they perceive it during the interaction.

Figure 7.2. Final Study Survey.
7.2 Experimental Results and Discussion

In order to evaluate our RL-based personalization approach, we compared both subjective measures (user survey) and objective measures (task performance) of the different conditions. More specifically, based on the survey results, we visualize the percentage of users who gave a high rating (points 4 and 5 in Likert scale) for the difficulty levels and robot feedback provided during each session, as well for robot intelligence. Moreover, we visualize total task performance for each user for both conditions. The results are shown in Figure 7.3.

Figure 7.3. Task Performance and User Survey Results.

Considering user survey results, we observe that 70% of the users agreed or totally agreed that the difficulty levels provided during the random condition were appropriate, while for the RL-based condition this percentage increases to 100%. However, for both conditions only 20% and 40% of users, for the random and the RL-
based condition respectively, thought that the robot provided appropriate feedback during the session. Moreover, only 20% of users thought that the robot was intelligent during the random condition. This percentage increases to 80% for the RL-based condition. Considering user comments, one of the users noted during the random condition that "the robot felt random during the session", while for the RL-based condition the same user commented that "the robot provided feedback when I needed it, increased my motivation and made me perform better". However, our proposed system provides a limited set of personalized policies which would not cover all different types of users, considering user skills, as well as preferences to different types of robot feedback. Considering robot feedback, further research is required to learn more advanced training strategies, considering robot behavior and verbal feedback [116].

As far as task performance is considered, we observe that 80% of the users achieved a higher score under the personalized condition. More specifically, the mean task performance was 26.5 for the personalized condition and 18.0 for the random one. Task performance is highly affected by the selected difficulty levels during the session, as dictated by the training policy. A basic assumption of our approach is that each user assessment model \(UM\) can be properly assigned to an existing cluster and its corresponding policy. A basic limitation of our approach arises when a user cannot be represented by an existing user model. Fitting a user into one of the existing models may not always fetch personalized results. A lack of exact skill-sets and preferences may result into a mismatched model and thereby in a failed personalization approach.

The KNN classification of the user assessment models \(UM\) assigned 20% of the participants to Cluster 1, 30% of the participants to Cluster 2 and 50% of the participants in Cluster 3, considering both sessions. Moreover, 50% of the participants were assigned to a different user model during the different conditions. Further data collection and analysis is needed to learn policies which can adequately handle
inconsistencies due to user model mismatch, unobserved user behaviors or changes in human behavior. To get further insights about the changes in the existing user models and their corresponding RL-based training policies, we visualize the root mean squared (RMS) differences between the assigned (prior) model and policy from the updated ones at the end of the session, as we show in Figure 7.4.

![Figure 7.4. RMS distances for user models and policies before and after the personalized training session.](image)

We observe that 70% of the users do not show significant changes in their performance, as depicted by the small RMS distances (RMS ≈ 0.1) of the user model vectors before and after the training session. Considering the training policies, we estimate the amount of change as $RMS(Q, Q')$ [31], where $Q$ is the initial Q-table (policy) and $Q'$ is the updated one, after the training session. Large distance values indicate that the applied policy changed significantly, as the system interacted with the user. Such changes may be due to mismatch between the user assessment model
and the assigned one. Moreover, changes in user’s behavior during the interaction may lead to inconsistencies between the assigned policy and an "optimal" one. A straightforward way to address such challenge would be to enable the system to switch between policies or re-assess the user, if a change in user behavior (performance and engagement) is detected.

To conclude with, we presented the design and evaluation of our proposed prototype system for SAR-based personalized cognitive training. Our system follows a human-guided assessment phase, where a human supervisor monitors and controls the assessment phase in order to assign each user to one of the existing user models. The system initializes its training policy by using the corresponding personalized policy. An interactive RL agent dynamically adjusts the assigned policy through Learning from Feedback (reward shaping). Our results indicate that our RL-based approach is a promising approach for real-time personalization. The analysis of both subjective and objective measures from the user study provided us with valuable insights and guidelines to formulate our future work and address the following challenges of such a data-driven approach: (1) how to utilize an existing set of user models and personalized policies by defining a parametric (generalized) mapping from user models to policies in order to address the challenge of outliers and rare user cases, (2) how to enable a robot to decide when to switch between policies or how to refine existing policies, considering changes in user behavior and performance during the interaction, in order to achieve a robust and efficient personalization, and (3) how to utilize Human-in-the-Loop and Interactive Machine Learning methods to learn more sophisticated personalized policies, from human demonstrations and guidance. Such policies may include strategies to maximize user performance (training), learning a new user or update knowledge for an existing one (assessment), as well as supportive behavior and motivational strategies, through verbal or non-verbal feedback.
CHAPTER 8
CONCLUDING REMARKS AND FUTURE DIRECTIONS

The research for this Thesis focuses on the aspect of personalization as a Machine Learning problem in the context of Robot-Assisted Training systems. We proposed a taxonomy in Robot-Assisted Training systems, highlighting the current research trends and needs towards developing a personalized Robot-Assisted Training system. Taking into consideration different personalization approaches, we formulated the problem of personalization as an interaction management problem; the problem of deciding what to do while interacting with a dynamic environment. We demonstrated how Reinforcement Learning can be used to solve such sequential decision making problems. Based on recent works in Robot-Assisted Training, Socially Assistive Robotics and Reinforcement Learning for HRI, we discussed how different types of users (primary, secondary) communicate with the system using different types of feedback (implicit, explicit, guidance) towards personalization. To this end, we proposed an Interactive Learning and Adaptation framework for Personalized Robot-Assisted Training which utilizes Interactive Reinforcement Learning methods to integrate different types of human-generated feedback to facilitate robot’s learning and provide the user with a personalized session.

As an experimental testbed, we designed a prototype SAR-based system for personalized cognitive training. We presented the Sequence Learning task; a cognitive task related to working memory and the ability of sequencing. We followed a data-driven approach towards developing a real-time system. We presented our data collection and analysis methods, as well as the user simulation approach to build a
set of user models and their corresponding personalized training policies. We showed how task engagement through EEG signals can be used to facilitate personalization through Learning from Feedback methods. Moreover, we proposed a Wizard-of-Oz system which utilizes informative interfaces and enables human supervisors to monitor and control the assessment phase of the Sequence Learning task. Finally, we concluded with a user study to evaluate the proposed prototype system. The experimental results show promising capabilities of our proposed framework and provided us with valuable insights towards refinement and improvement.

A main limitation of our approach is the assumption that the proposed SAR-based system is able to provide information about performance considering the sole feature of EEG-based task engagement. However, EEG signals can be very noisy and a more robust analysis should be investigated. Moreover, there are also other critical parameters which should be considered in robot adaptation. For example, a robot can read comfort/discomfort signals from human users towards adaptation [130]. Similarly, fatigue and frustration, as well as aspects of social interaction [89], and features like response and completion time are parameters which should be accounted for [19]. Moreover, one of the problems with data-driven approaches is that the system learning gets adapted to limited data and user types, thereby not being able to address issues when interacting with "rare" user cases.

The long-term goal of this research is to design and develop a unified personalization framework for robot-based assessment and training, which considers training to be a collaborative approach in which both human and robot participate in identifying the best personalization policy. In order to adjust to human preferences, training needs, abilities in an interactive manner, the system must enable and utilize the interaction between its components. The development of such intelligent sys-
tems requires computationally advanced mechanisms, to ensure a safe and efficient interaction between the robotic system and its physical environment.

To address such challenges, an appropriate approach would be to model the system as a Cyber-Physical System. A Cyber-Physical System (CPS) is a smart, closed-feedback system, in which the computational and physical components are integrated to control and sense the changing state of real-world variables [131]. Advances in CPS include novel approaches for personalization and progressive co-adaptation [77]. Recent works in CPS highlight the role of human users in shaping system’s behavior and interaction, suggesting the integration of the human component, as an additional component in a CPS system [132]. Following this approach, we propose a Human-centric Cyber-Physical System (HCPS), which enables the human component (different types of users) to communicate with the other system components, towards a safe and personalized HRI [133, 134].

![HCPS system architecture](image)

Figure 8.1. HCPS system architecture.
In order to illustrate the proposed HCPS architecture, future work includes the development of an interrelated set of robot-based tasks for cognitive assessment and training. In the context of vocational training, cognitive assessment is important when dealing with workplace automation, where a worker must remain attentive, able to handle high cognitive load, switch tasks/materials on call, and rely on good working memory to avoid mishaps. As an ongoing work, we develop task prototypes as variations and improvements of the sequence learning task. To investigate the effect of the training environment to user’s performance, we propose to build both virtual and physical task environments [135], as we show in Figure 8.2.

Figure 8.2. Future and ongoing work on developing an experimental testbed for vocational cognitive assessment and training.

The different tasks will be designed in order to assess and train specific cognitive aspects (e.g., working memory, cognitive flexibility, etc.), following guidelines for
well-established measures related to cognitive, emotional, sensory and motor functions \(^1\). We follow a *cyberlearning* approach, to highlight the need of information sharing and communication between the system and the potential users, experts and stakeholders [136]. The proposed system includes an assessment phase, where the system assesses different aspects of cognitive skills through multisensing assessment tasks. This information is stored and analyzed in order to represent and detect individual differences and weaknesses in cognitive functioning. This analysis can then be used to recommend a specific set of training tasks to provide personalized and adaptive training. A human supervisor can monitor and guide the system through all phases, as we show in Figure 8.3, in order to facilitate personalization and safe interaction, across all different phases.

![Figure 8.3. Vocational Assessment and Training.](http://www.healthmeasures.net/explore-measurement-systems/nih-toolbox)
We are interested in investigating how different cognitive and emotion factors interrelate with each other to impact human performance and learning, focusing on HRI-based training and educational systems. Research in computational modeling in cognition and emotion try to explain emotion in the context of its relationship with cognition [137, 138]. Computational approaches to design and test such models and architectures include the ACT-R and SOAR architectures, which provide models for problem solving, decision making, working memory, and others [139].

Figure 8.4. Cognitive and emotion models can be utilized to inform personalized robot-assisted assessment and training systems. Moreover, such systems can be used as experimental testbeds to study existing cognitive and affective architectures.

For the purposes of HRI, such an architecture can provide a robot with a user model, as an indication of the mental and affective state of the user, during an assessment or training task. On the other hand, robot-based systems can be used as testing platforms to develop and evaluate theory-based computational models and architectures for emotion and cognition. In the context of Human-Robot Collaboration, shared mental models have been proposed to promote effective human-robot teaming
and training [140]. In the context of Affective Computing [102], recent studies investigate the role of emotion in Reinforcement Learning robots and agents, focusing on how computational emotion models affect agent’s learning behavior and performance [141]. Cognitive models have also been developed and integrated to computer-based training systems for dyslexia assessment [142].

Robotic systems have been successfully deployed as a new intervention tool to augment therapy and training. Robotic psychology is an emerging research area which studies (1) how robots can be used in psychology (e.g., evidence-based therapy, psychological assessment) and (2) how psychology and cognitive sciences can provide guidelines for more effective human-robot interactions and intervention tools [143]. As a future research direction, we are interested in investigating how cognitive and emotional processes affect (and are affected by) human performance and learning, during robot-based assessment and training systems, with applications in education, healthcare and manufacturing.
REFERENCES


reinforcement learning for interactive systems and robots,” ACM Transactions

[66] A. Zehfroosh, E. Kokkon, H. G. Tanner, and J. Heinz, “Learning models of
human-robot interaction from small data,” in Control and Automation (MED),

reinforcement learning?” in EWRL 2015: Proceedings of the 12th European
Workshop on Reinforcement Learning, 2015.


[69] P. Abbeel and A. Y. Ng, “Apprenticeship learning via inverse reinforcement
learning,” in Proceedings of the twenty-first international conference on Machine

[70] E. Mikołajewska and D. Mikołajewski, “Neurological telerehabilitation-current

using virtual reality task can improve balance in patients with stroke,” Disability

“Design of a home-based adaptive mixed reality rehabilitation system for stroke
survivors;” in Engineering in Medicine and Biology Society, EMBC, 2011 An-


BIOGRAPHICAL STATEMENT

Konstantinos Tsiakas was born in Athens, Greece. In 2012, he received his Diploma of Engineering (Dipl. Eng.) from Electronic and Computer Engineering Department of Technical University of Crete, Greece. In his Diploma Thesis he conducted research on continuous-space Language Modeling using Gaussian Mixture Models, supervised by Prof. Vasilis Digalakis.

As a researcher of HERACLEIA lab, at the University of Texas at Arlington, he participated as a graduate research assistant in several NSF-funded projects, under the supervision of Prof. Fillia Makedon and he defended his PhD in March 2018. During his PhD, he was also a research fellow for the National Center for Scientific Research - Demokritos, under the supervision of Prof. Vangelis Karkaletsis.

His research interests revolve around Artificial Intelligence, focusing on Reinforcement Learning and Interactive Machine Learning, Human-Computer and Human-Robot Interaction with applications to Robot-Assisted Training and Socially-Assistive Robotics. Konstantinos is currently a visiting research faculty at HERACLEIA lab and will also be a post-doctoral scholar at Yale Department of Psychiatry, under the supervision of Prof. Morris Bell, where he will be involved in developing Machine Learning methods for improving assessment of executive function in children.