SIKS Dissertation Series No. 2023-28
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

ISBN: 978-94-6483-462-8

Typeset by: LaTeX.
Printed by: Ridderprint
Cover design by: Midjourney (2023) & Marilou Maes, persoonlijkproefschrift.nl

VRIJE UNIVERSITEIT

LEARNING TO BEHAVE

Reinforcement Learning in Human Contexts

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad Doctor of Philosophy aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. J.J.G. Geurts,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de Faculteit der Bètawetenschappen
op dinsdag 14 november 2023 om 9.45 uur
in een bijeenkomst van de universiteit,
De Boelelaan 1105

door

Floris den Hengst

geboren te Kampen

promotoren:	prof.dr. F.A.H. van Harmelen prof.dr. M. Hoogendoorn
copromotor:	dr. V.G.M. Francois Lavet
promotiecommissie:	prof.dr. A.C.M. ten Teije prof.dr. M. Dastani prof.dr. A. Plaat dr. M. van Otterlo prof.dr. P. Traverso

Preface

De totstandkoming van dit proefschrift zou onmogelijk zijn zonder Mark Hoogendoorn, omdat je mij ruim zes jaar geleden het vertrouwen gaf om deze uitdaging aan te gaan en mij hierna steunde met een schijnbaar oneindige hoeveelheid positieve energie en ontelbare goede adviezen. Ook tegenover Frank van Harmelen is mijn dank groot, voor alle kritische noten en een blik die waar nodig weids overziet maar ook feilloos kan scherpstellen op cruciale details. Ook Vincent François-Lavet wil ik bedanken: als latere toevoeging aan het begeleidingsteam waren er van jou hand meteen goede adviezen en met de RL summer school gaf je een grote impuls aan mijn enthousiasme voor het onderzoeksveld. Daarnaast wil ik Joost Bosman bedanken voor het in mij gestelde vertrouwen. Je blik op de rol van wetenschap en techniek binnen een grote en maatschappelijk ingebedde organisatie als ING was verfrissend en zal me altijd bij blijven.

I would like to thank the members of the committee for taking interest in this thesis, for taking the time to read it, and for their questions during the upcoming defense. Additionally, I am grateful for all of the mostly anonymous reviewers for shaping this thesis with criticisms, questions, and, of course, suggestions for the inclusion of related work.

Special thanks go out to coauthors of the papers included in this thesis. Eoin, Ali, Sandjai, Ehsan, Yannick, Paul, and Martijn, I have deeply enjoyed getting to know you, collaborating with you, and learning from all of you.

Vervolges wil ik Kylie bedanken, omdat ze ondanks de weekendjes type-werk en de nodige frustratie na een mislukt experiment mijn vrouw wilde worden. Kylie, je liefde en vertrouwen zijn een bron van licht die me door de donkerdere perioden van dit onderzoeksavontuur heen hebben geholpen.

Mijn dank is ook groot voor mijn naaste familie Hugo, Judith & Eric, Daniël & Marina, Lizzy en Daniëlla en in het bijzonder mijn ouders Jan en Ineke voor jullie steun, enthousiasme en interesse in mijn onderzoek.

I had the pleasure of working with great people in the CI and QDA groups. My gratitude goes out to Milan, Gongjin, Karine, Diederik, Jacqueline, Luca, Bart, Ward, Frank, Lucas, David, Luis, Alessandro, Anne, Fuda, Jan, Tariq, Guszti, Yvonne, Jacob, Louk, Shujian, Arwin, Olivier, Joshua, Buelent, William, Martijn, Evert, Matteo, Steffen and Emile for all the nice rooftop lunches, borrels, parties, COVID team outings at home, coffee machine talks, movies, and of course the traditional Christmas/NY dinner parties.

I am thankful also for all the help and shared experience in putting research and technology into daily practice at ING. I would never have been able to succeed over these past years without Roel, Elvan, Patrick, Joost jr., Bas, Maartje, Hadi, Wout, Anton, Pinar, Ralf, Arie, Marzieh, Evert-Jan, Luis, Sara, George, Leonhard, Lorena, and Eileen.

Tot slot wil ik mijn vrienden in het algemeen, en de vrienden van de 'Oud&Nieuw' groep in het bijzonder, bedanken voor alle interesse en het nodige geduld wanneer onderzoek wat nog in ontwikkeling was ter sprake kwam. De ongedwongen gesprekken die daardoor ontstonden hebben me telkens weer met frisse ogen naar mijn eigen onderzoek doen kijken en hebben me daarmee tot allerlei nieuwe ideeën en vragen geleid die uiteindelijk weer in dit proefschrift terecht zijn gekomen.

Contents

Publications

[P1] F. den Hengst, V. François-Lavet, M. Hoogendoorn and F. van Harmelen. 'Planning for potential: efficient safe reinforcement learning'. In: *Machine Learning* (2022), pages 1–20.

[P2] F. den Hengst, M. Otten, P. Elbers, V. François-Lavet, M. Hoogendoorn and F. van Harmelen. 'Guideline-informed reinforcement learning for mechanical ventilation in critical care'. In: *Artificial Intelligence In Medicine* (in submission).

[P3] F. den Hengst, V. François-Lavet, M. Hoogendoorn and F. van Harmelen. 'Reinforcement Learning with Option Machines'. In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. Edited by L. D. Raedt. Main Track. International Joint Conferences on Artificial Intelligence Organization, July 2022, pages 2909–2915.

[P4] F. den Hengst, E. M. Grua, A. el Hassouni and M. Hoogendoorn. 'Reinforcement learning for personalization: A systematic literature review'. In: *Data Science* 3.1 (2020), pages 107–147.

[P5] F. den Hengst, M. Hoogendoorn, F. van Harmelen and J. Bosman. 'Reinforcement Learning for Personalized Dialogue Management'. In: *2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. ACM. 2019, pages 59–67.

[P6] Y. Smit, F. den Hengst, S. Bhulai and E. Mehdad. 'Strategic Workforce Planning with Deep Reinforcement Learning'. In: *International Conference on Machine Learning, Optimization, and Data Science*. Springer, 2022.

[P7] M. van Zeelt, F. den Hengst and S. H. Hashemi. 'Collecting High-Quality Dialogue User Satisfaction Ratings with Third-Party Annotators'. In: *Proceedings of the 2020 Conference on Human Information Interaction and Retrieval*. CHIIR '20. Vancouver BC, Canada: Association for Computing Machinery, 2020, pages 363–367.

Introduction

1 Introduction

Since the start of Artificial Intelligence (AI) as a field of study during the Dartmouth Summer Research Project on Artificial Intelligence, one of its key goals has been to create machines that "exhibit every aspect of learning" and that "improve themselves" [230]. Over the decades since, the preconditions for developing such machines such as large amounts of available data, proximity to and capabilities of electronic devices are increasingly being met. As a result, the field that takes up the study and practice of adaptive machines, commonly referred to as *machine learning*, has established itself as a serious academic effort as well as a significant industry globally.

The growing interest in and impact of learning systems is accompanied by a growing skepticism of – or even animosity to – these technologies. Issues such as diminishing privacy, a (perceived) lack of control and lacking safety guarantees are often cited [6, 86, 303]. These worries have found their way to the societal and academic agendas, as reflected by various initiatives to inform the wider society and efforts to develop controllable, interpretable and safe adaptive systems [88, 423, 426, 370].

Reinforcement learning (RL) is a particularly general framework to AI, aimed at obtaining behaviors that achieve the best expected outcome. The RL framework is about "learning what to do—how to map situations to actions—so as to maximize a numerical reward signal" [338]. In RL, actions are taken by an agent in a sequence of situations known as states of its environment. In some cases, the agent knows everything about the environment: its possible states, the effects of actions and the associated rewards. In these cases, the agent can find a suitable solution by reasoning and planning, i.e. performing operations on a representation of the environment internal to the agent. Challenges still exist here, for example state and action spaces that are too large to represent internally or reason over [322]. In other cases, not everything about the environment is known. In these cases, the agent has to attempt various actions in the environment and observe their results in order to achieve its goal.

In the past decades, RL has shown many impressive real-world applica-

tions in its relatively brief history. An early real-world success is commonly recognized in Tesauro's backgammon player that learned to play at a "strong intermediate level" with solely a set of randomly initialized neural networks weights and a sufficient number of games played against itself [351]. Its recent counterpart can be found in AlphaGo Zero by Silver et al. [322], which defeated the best human player in the game of Go in essentially the same way. Within e-sports, RL has outperformed top human champions in the individual setting of a car racing game [392], as well as the team setting of a real-time strategy game [374]. Within computer hardware design, RL is used to generate manufacturable chip floorplans in under six hours. A stark contrast to the "months of intense effort by physical design engineers" otherwise required for this challenging problem [237]. Within continuous control, RL has been used to control tokamak plasmas for nuclear fusion-based power generation [78], to control stratospheric balloons that bring internet access to remote areas in highly unpredictable conditions due to e.g. wind speed [26] and to minimize energy usage for cooling data centers [189].

These successes indicate the power and potential for RL to improve our lives. Many of the successful applications of RL so far, however, seem to have been found in highly controlled settings that involve humans only to a very limited extent. In many of the previously mentioned successes of RL, humans are effectively entirely absent or the interaction between agent and human is confined within the rules of a game. Such interaction patterns are not present in important human-centered domains such as healthcare, finance, robotics, education, government and retail. In order to investigate and increase the potential for impact of RL, therefore, we have to study it in contexts that involve humans. This is a broad challenge that, among others, includes the interlinked issues of data efficiency, safety, and controllability [86]. We continue this section by detailing each of these challenges and their interactions.

Data Efficiency

Adaptive systems that interact with humans generally have to be data efficient. Firstly, every sub-optimal interaction between human user and agent comes at the inherent cost of human patience. The user may simply lose interest and forgo further interactions if performance is not up to the users' standards for extended periods of time. To illustrate, the previously mentioned AlphaGoZero required 4.9 million games to achieve *master* status in the game of Go [424]. Secondly, there may be legal and ethical objections to subjecting users to more sub-optimal behavior than strictly necessary. Think, for example, of the healthcare setting, where incorrect behaviors directly impact human life and well-being. This example highlights the relation between data efficiency and safety. If data collection comes at the risk of unsafe behavior, then increasing data efficiency automatically results in increased safety. On top of this, data efficiency is related to controllability: the ability to control the agent to some degree may eliminate the need for experimentation by the agent. Data efficiency is, finally, of particular importance in human contexts because

of the typical absence of suitable simulators. Human behaviors, preferences, mental states, physical states, etc. are notoriously heterogeneous, challenging to model and hard to predict. Training an agent in interaction with a simulator may therefore not be feasible while obtaining these data in large volumes from real world situations is typically prohibitively expensive, both financially and in terms of human effort.

Safety

All of the arguably most important and high-impact contexts, such as healthcare, finance, education, robotics in human contexts, etc. come with safety constraints that have to be met at all times [422, 192]. While we humans care about these contexts to such a degree that we have agreed to regulate them, RL does not support such safety constraints out of the box [6, 86, 114]. Additional systems have to be put in place to guarantee that humans are safe when interacting with RL agents. While safety is a significant niche within RL research, modelling and enforcing safety constraints remain challenges in many realistic settings [86, 114, 264]: what formalisms are suitable for defining safety constraints such that they are powerful enough to express real-world constraints and descriptive enough for domain experts to verify their correctness? How do these safety mechanisms impact the learner: do constraints limit the solution space and always reduce the learning problem as a result [5]? When are constraints so restrictive that finding an acceptable solution via trial-and-error alone becomes infeasible [184]? And if constraints negatively impact the learner, what can we do to mitigate this impact?

As we mentioned previously, safety is related to data efficiency: an agent that learns quickly, may make less critical errors. On the other hand, safety constraints may both result in a more complex task and inhibit exploration, and decrease data efficiency as a result. Additionally, ensuring the safety of an RL agent can be viewed as the ability to prevent the agent from unsafe behaviors. Safety, in this sense, can be seen as a special case of controllability as well.

Controllability

Traditional RL solely relies on a reward function to quantify the appropriateness of agent behavior. While the reward function is a very general mechanism [323], it may not be sufficient in scenarios where agents interact with human users. These users may have some notion of what a good solution looks like. In this case, the agent should be able to benefit from this knowledge, e.g. to stay safe during learning or to learn more data efficiently. A popular practice for communicating such pre-existing knowledge to the agent is *reward shaping* [254]. Additional rewards are given in hopes of guiding the agent to the desired behavior. While reward shaping is useful, it is generally done in an ad-hoc and problem-specific way and can be tedious. An alternative approach known as curriculum learning comes with similar issues. In curriculum learn-

ing, a series of tasks is formulated and presented to the agent in increasing complexity in order speed up or enable learning of a final complex task [29]. Alternatively, users may control the agent directly in order to demonstrate what suitable behavior looks like [150]. Generating these demonstrations, however, typically requires proficiency at controlling the agent.

Another issue of controllability is that we may want to give a single agent different goals at different times. Achieving such dynamic goals is certainly possible with regular RL by e.g. including the goal in the state representation. Regular RL however, does not take into account the special structure of dynamic goals in RL and as a result may have to relearn from scratch when given a new goal that is only slightly different from a previous goal. Approaches exist for reusing previously learned knowledge, including goal-conditioned RL [170, 311] and hierarchical RL [325, 335]. Such an approach, however, generally requires that the goals and structure of the task is expressed fully up-front. This is typically not the case with instructions that people give each other, whether these are of an informal nature, such as recipes in cooking and directions in navigation, or more formal such as medical guidelines or financial regulations. In order to increase the potential of RL in human context, it has to be be able to benefit from incomplete instructions as instructions that people tend to give are incomplete.

1.1 Research Questions

This thesis aims to contribute to the development and usage of RL techniques in human contexts. In order to break down the overarching challenge of developing and using RL techniques in human contexts, we address five particular research questions (RQS):

RQ1a How has RL been applied to personalization?

RQ1b How can we improve and personalize the decision-making in dialogue agents with RL?

RQ1c How can we improve decision-making with RL when end-users and agents interact in terms of goals and solutions?

RQ2 How do safety constraints affect RL learning tasks and how can we improve data efficiency of safe RL?

RQ3 How can we control RL agents to improve safety and data efficiency?

RQs 1a-1c aim to support how RL *has been* and *can be* used in contexts involving humans. RL agents can interact with human end-users either directly or indirectly. In the former pattern, a (group of) human user(s) constitutes the agent's environment which the agent can affect through its actions. We hereby refer to this interaction pattern as *personalization*. A comprehensive overview of RL usages for personalization is currently lacking. Such an overview would

inform us about how the aforementioned challenges are tackled in practice in this interaction pattern. In the latter, indirect, interaction pattern, the user does not constitute the environment but provides the goals for the agent to achieve and the constraints under which to achieve these. The agent learns a solution and then presents this solution to the user in this interaction pattern. We study both of these interaction patterns to answer research questions RQ1a-RQ1c.

Having built an understanding of the usage of RL in human contexts, we can turn to the challenges listed above with RQ2 and RQ3. The first of these addresses the combined challenge of safety and data efficiency and is motivated by the gaps in our understanding on the interplay between strict safety constraints and data efficiency identified above. The second research question deals with ways of controlling the agent and it relates to both the challenges identified earlier in the sections 'Safety' and 'Controllability'. Regarding the challenge of safety, it is particularly interesting *how* safety constraints can be expressed so that both these both useful for both the agent and a human (expert) . In this thesis we consider the usage of linear temporal logic for controlling the agent both in terms of safety and data efficiency [275]. This logic has recently become of interest to the RL community due to its ability for expressing formulae about the future of paths, which is helpful when expressing what constitutes a *safe* path [5, 108, 169, 184, 385, 398]. At the same time, these expressions can be used to break down a full task into smaller subtasks. Solutions to subtasks may be easier to find and additionally, may be combined to solve previously unseen tasks [9, 48, 157, 158].

1.2 Scope

This thesis is composed of a set of papers written during a period of five years. The papers are each presented in a separate chapter and all received slight editorializing for inclusion in this work. Specifically, tables and figures were resized, notation and terminology was harmonized and citations of own works were updated to refer to chapters. The set of papers collectively address a set of topics that in turn form the scope of this thesis. These topics are:

1. Reinforcement Learning for Personalization in Chapters 2, 3 and 6
2. Adaptive dialogue agents in Chapters 3 and 4
3. Operations Management in Human Context in Chapter 5
4. Safe reinforcement learning in Chapters 6 and 7
5. Reinforcement learning with instructions in Chapter 8

The first topic answers research questions 1a and 1b. The second topic deals with research question 1c. The third topic deals with research 2 and the final topic deals with research question 3. We divide these topics into two parts:

Part	Chapter	Paper	RQ
I	2	den Hengst et al. [P4]	1a
	3	den Hengst et al. [P5]	1b
	4	van Zeelt, den Hengst and Hashemi [P7]	1b
	5	Smit et al. [P6]	1c
II	6	den Hengst et al. [P2]	2
	7	den Hengst et al. [P1]	2,3
	8	den Hengst et al. [P3]	3

Table 1.1: Overview of parts, chapters, papers and research questions addressed in this thesis.

Part I outlines innovations where RL brings benefits when applied in a human context and Part II contains theoretical and algorithmic advances.

Table 1.1 displays the structure of this thesis by related topics, research questions and papers. By studying various applications of RL within human contexts and proposing several improvements to the field of RL, we increase its potential for impact in human contexts. We continue by elaborating on the contributions of this thesis to these topics.

1.2.1 Reinforcement Learning for Personalization

We performed a systematic literature review into the usage of RL for personalization. In this Chapter, we first describe how RL can be used for personalization, and then introduce a framework to categorize related work. This framework consists of a description of the settings in which RL has been used for personalization, a set of aspects to describe the particular solutions developed and a characterization of the evaluation used. We then categorize related work using this framework and identify trends, challenges and opportunities for future work. Our primary finding is a marked increase in the number of studies that use RL for personalization over time. This increase, however, is not mirrored by a comparable increase in studies that evaluate in a 'live' setting. The framework and categorization additionally allow researchers and practitioners to quickly navigate the field and identify relevant related work. The systematic literature review is included in Chapter 2 of this thesis.

Next, we proposed to use RL for personalization of a dialogue agent in order to increase its performance. We demonstrated that RL can be used to personalize the dialogue management module of dialogue agents and that this can lead to an improved performance over an existing manually constructed gold standard. In particular, we found that RL can tailor the decision-making of a recommendation agent across domains and that the RL-driven approaches best transfer to the newly introduced domain of financial product recommendation. In this application, an RL agent directly interacts with a human. The paper describing these efforts and outcomes can be found in Chapter 3.

Finally, we create a safe RL approach in the medical domain. The approach is used to optimize mechanical ventilation settings in critical care. Learned policies make decisions based on characteristics of the individual patients, including demographic and physiological variables. In our evaluations, we found that policies trained with RL selected more varied actions than those made by clinicians. Additionally, we found that policies could be learned that select actions compliant to a medical guideline on protective lung ventilation at limited cost to overall performance. In this application, an RL agent can interact directly or indirectly with a human: either by selecting ventilator settings autonomously or by advising clinicians on favourable settings. This approach to safe RL in medicine is described in Chapter 6.

1.2.2 Adaptive dialogue agents

Communicating with artificial agents in a conversational interaction style has been one of the central challenges in Artificial Intelligence since its initiation as a field of research [230, 366]. A crucial component of this challenge is including personal context in the dialogue, since personal context is crucial to communication between humans [30]. We describe an approach to personalizing dialogue agents. We additionally include novel comparisons of RL-based approaches to a state-of-the-art approach based on entropy minimization and to a heuristics-driven approach. These efforts are described together with our efforts into personalizing dialogue agents in Chapter 3.

Additionally, we have investigated the evaluation of dialogue agents. A key challenge here is to evaluate whether a dialogue has met the users' information needs. We have little opportunity to ask the user when they have aborted the interaction and typically do not know whether they did so because of being satisfied with the information given so far or rather because they lost patience and continue by obtaining the desired information by some other means. In Chapter 4 we list best practices for collecting annotations of using third-party annotators, i.e. judges of dialogue quality that are not using the system, and we introduce a tool for collecting annotations that implement these.

We additionally build on this topic by using it as an application area in the study of safe RL as described below.

1.2.3 Operations Management in Human Context

We have studied the use of RL in an important problem in operations management: strategic workforce planning (SWP). This problem is not only common and challenging, it is also provides an interesting test-bed for applying RL in human context. Firstly, because a SWP decision support tool makes decisions about people in an organization. This will require a proper alignment between the formal optimization objectives and the actual intended outcome by domain experts using the tool. SWP is, secondly, a complementary test-bed to those of personalization and dialogue agents in the way the user interacts with the agent. In particular, the user interacts directly, i.e. at the level of states and

actions, with the agent in the personalization and dialogue agent-settings. In contrast, a second and indirect level of interaction is present for SWP. The user interacts with the agent at the level of rewards and policies in the sense that the user specifies the reward and inspects the agents' policy in SWP.

To tackle this SWP problem, we contributed a simulation-optimization approach and have studied its applicability and performance in a comparison with a linear-programming baseline in Chapter 5. In particular, we compare two common scenarios of SWP and compare performance between the baseline and proposed approach. The scenarios differ in how the optimization goal is defined: in the first scenario, the goal is easy to optimize for with existing approaches but hard to express for domain experts whereas the goal is easy to express in the second scenario. We find that the proposed approach performs comparable to the baseline in the first scenario while it outperforms the baseline in the second scenario.

1.2.4 Safe Reinforcement Learning

We have proposed a framework in which statements in medical guidelines are operationalized as safety constraints in a RL learning process. We evaluated the approach in a study with observational data and results indicate that our approach has the potential to decrease 90-day mortality while ensuring guideline adherence. Since the learned policies come with safety guarantees, they may be more trusted by clinicians relying on the policies' decision-making. These contributions may be found in Chapter 6.

We have additionally studied how safety constraints can be modeled in linear temporal logic (LTL), a formalism that was designed to model computer programs and in which properties that include a notion of time can be expressed symbolically. We analyzed how safety constraints impact expected future rewards and showed a relation between expected rewards and the progress toward a goal in a particular representation of the safety constraints known under the umbrella term of *automata*.

We have then proposed an algorithm to scale safe RL with constraint complexity based on symbolic reasoning, i.e. planning. We use planning to infer progress toward a symbolically expressed goal and then inform the learner of progress using potential-based reward shaping in this algorithm. In doing so, we do not only use the symbolic constraints to limit the learner, but also leverage these to guide it. We have modeled real-world constraints from the banking domain and applied them to the adaptive dialogue agent case study introduced in Chapter 3. Our efforts related to safety in RL can be found in Chapter 7.

1.2.5 Reinforcement Learning with Instructions

We have proposed a framework for using high-level instructions within RL. Of note is that the framework can benefit from instructions that are incomplete, making it widely applicable. It includes a way to encode such instructions, to control an agent with such instructions and to learn from experiences. The

framework enables the learning of named behaviors using only incomplete instructions and experiences. The learned behaviors are associated with names from the instructions. Humans can choose names that are meaningful to them, such as 'collect wood' or 'move north'. These behaviors can therefore easily be reused in instructions for other tasks. In evaluations of the framework, we show that it outperforms the state-of-the-art in single-task, multi-task and zero-shot settings.

After our earlier contributions on safe RL, learning with instructions is a second approach to controlling RL agents. These approaches are quite different when viewed up close: our contribution on safe RL is specific to a setting with safety constraints and alters the reward function whereas our contribution on RL with instructions applies broadly and uses hierarchical RL. There is however, an important similarity to both contributions. Both rely on the usage of automata and temporal logic to encode prior knowledge of either safety constraints or task instructions. Our contributions to the topic of RL with instructions were inspired by our insights on safe RL with automata and can be found in Chapter 8.

1.3 Overview and Personal Contributions

The personal contributions to the papers in this thesis made by the author are:

Chapter 2: den Hengst et al. [P4] I initiated and took the lead in this study, proposed the framework used for categorization, took the lead in data gathering and served as corresponding author. In equal participation with Ali el Hassouni and Eoin Grua, I worked on the data analysis and writing of several sections of this work.

Chapter 3: den Hengst et al. [P5] I acted as lead in this research and was involved in all of its components. I proposed the study design and methodology, including all novel approaches included therein. I implemented these approaches, ran the experiments, analyzed their results and created figures and tables to present these. I wrote the paper to present these.

Chapter 4: van Zeelt, den Hengst and Hashemi [P7] Together with Mickey van Zeelt, a M.Sc. student in Information Systems who I co-supervised with dr. Seyyed Hadi Hashemi, I was actively involved in all stages of research. In particular, I participated in the survey of literature, the study design and data collection. I took the lead in data analysis and writing of the paper and serve as second author of this paper.

Chapter 5: Smit et al. [P6] I developed this work together with Yannick Smit, a M.Sc. student in Stochastics and Financial Mathematics for whom I acted as daily supervisor, and dr. Ehsan Mehdad. While Ehsan first proposed the project, I worked on the motivation, conceptualisation

and literature study phases of this work. Together with Yannick, I developed the proposed approach and the experimental setup. I reviewed the implementation by Yannick. I participated in the data analysis, the generation of figures and I wrote the paper as joint first author.

Chapter 6: den Hengst et al. [P2] I initiated the project, contributed the framework, developed the code for RL modeling, RL training and evaluation. The guideline encoding was created in consultation with co-authors Martijn Otten and Paul Elbers and Haritha Jayaraman for her Msc. thesis project in a related project in which I acted as daily supervisor. I serve as first author for this paper.

Chapter 8: den Hengst et al. [P3] I took the lead in this research and contributed the approach, experimental setup and implementation. I conducted the experiments, analyzed the results and wrote the paper.

Chapter 7: den Hengst et al. [P1] In this project, I took on the role of project lead. I performed the theoretical analysis and proposed the algorithm based on this analysis. I designed the experimental setup, implemented it and conducted the experiments. I analyzed the results, generated figures and tables to present these and wrote the paper included in this thesis.

Part I

Applications of Reinforcement Learning in Human Contexts

2

RL for Personalization: A Systematic Literature Review

The major application areas of reinforcement learning (RL) have traditionally been game playing and continuous control. In recent years, however, RL has been increasingly applied in systems that interact with humans. RL can personalize digital systems to make them more relevant to individual users. Challenges in personalization settings may be different from challenges found in traditional application areas of RL. An overview of work that uses RL for personalization, however, is lacking. In this work, we introduce a framework of personalization settings and use it in a systematic literature review. Besides problem setting, we review solutions and evaluation strategies. Results show that RL has been increasingly applied to personalization problems and realistic evaluations have become more prevalent. RL has become sufficiently robust to apply in contexts that involve humans and the field as a whole is growing. However, it seems not to be maturing: the ratios of studies that include a comparison or a realistic evaluation are not showing upward trends and the vast majority of algorithms are used only once. This review can be used to find related work across domains, provides insights into the state of the field and identifies opportunities for future work.

Based on [P4]:
Floris den Hengst, Eoin Grua, Eoin Martino Grua, Ali el Hassouni and Mark Hoogendoorn
Reinforcement Learning for Personalization: A Systematic Literature Review
Data Science 2020

2.1 Introduction

For several decades, both academia and commerce have sought to develop tailored products and services at low cost in various application domains. These reach far and wide, including medicine [12, 123], human-computer interaction [102, 215], product, news, music and video recommendations [291, 293, 381] and even manufacturing [67, 272]. When products and services are adapted to individual tastes, they become more appealing, desirable, informative, e.g. *relevant* to the intended user than one-size-fits all alternatives. Such adaptation is referred to as *personalization* [93].

Digital systems enable personalization on a grand scale. The key enabler is data. While the software on these systems is identical for all users, the behavior of these systems can be tailored based on experiences with individual users. For example, Netflix's[1] digital video delivery mechanism includes tracking of views and ratings. These ease the gratification of diverse entertainment needs as they enable Netflix to offer instantaneous personalized content recommendations. The ability to adapt system behavior to individual tastes is becoming increasingly valuable as digital systems permeate our society.

Recently, reinforcement learning (RL) has been attracting substantial attention as an elegant paradigm for personalization based on data. For any particular environment or user state, this technique strives to determine the sequence of actions to maximize a reward. These actions are not necessarily selected to yield the highest reward *now*, but are typically selected to achieve a high reward in the long term. Returning to the Netflix example, the company may not be interested in having a user watch a single recommended video instantly, but rather aim for users to prolong their subscription after having enjoyed many recommended videos. Besides the focus on long-term goals in RL, rewards can be formulated in terms of user feedback so that no explicit definition of desired behavior is required [23, 139].

RL has seen successful applications to personalization in a wide variety of domains. Some of the earliest work, such as [315], [316] and [408] focused on web services. More recently, [197] showed that adding personalization to an existing online news recommendation engine increased click-through rates by 12.5%. Applications are not limited to web services, however. As an example from the health domain, [415] achieve optimal per-patient treatment plans to address advanced metastatic stage IIIB/IV non-small cell lung cancer in simulation. They state that 'there is significant potential of the proposed methodology for developing personalized treatment strategies in other cancers, in cystic fibrosis, and in other life-threatening diseases'. An early example of tailoring intelligent tutor behavior using RL can be found in [221]. A more recent example in this domain, [129], compared the effect of personalized and non-personalized affective feedback in language learning with a social robot for children and found that personalization significantly impacts psychological valence.

Although the aforementioned applications span various domains, they are

[1] `https://www.netflix.com`

similar in solution: they all use traits of users to achieve personalization, and all rely on implicit feedback from users. Furthermore, the use of RL in contexts that involve humans poses challenges unique to this setting. In traditional RL subfields such as game-playing and robotics, for example, simulators can be used for rapid prototyping and *in-silico* benchmarks are well established [27, 39, 85, 180]. Contexts with humans, however, may be much harder to simulate and the deployment of autonomous agents in these contexts may come with different concerns regarding for example safety. When using RL for a personalization problem, similar issues may arise across different application domains. An overview of RL for personalization across domains, however, is lacking. We believe this is not to be attributed to fundamental differences in setting, solution or methodology, but stems from application domains working in isolation for cultural and historical reasons.

This paper provides an overview and categorization of RL applications for personalization across a variety of application domains. It thus aids researchers and practictioners in identifying related work relevant to a specific personalization setting, promotes the understanding of how RL is used for personalization and identifies challenges across domains. We first provide a brief introduction of the RL framework and formally introduce how it can be used for personalization. We then present a framework for characterizing problem settings. The purpose of this framework is for researchers with a specific setting to identify relevant related work across domains. We then use this framework in a systematic literature review (SLR). We investigate in which settings RL is used, which solutions are common and how they are evaluated: Section 2.5 details the SLR protocol, results and analysis are described in Section 2.6. All data collected has been made available digitally [80]. Finally, we conclude with current trends challenges in Section 2.7.

2.2 Reinforcement learning for personalization

RL considers problems in the framework of *Markov decision processes* or MDPs. In this framework, an agent collects rewards over time by performing actions in an environment as depicted in Figure 2.1. The goal of the agent is to maximize the total amount of collected rewards over time. In this section, we formally introduce the core concepts of MDPs and RL and include some strategies to personalization without aiming to provide an in depth introduction to RL. Following [338], we consider the related *multi-armed* and *contextual bandit* problems as special cases of the full RL problem where actions do not affect the environment and where observations of the environment are absent or present respectively. We refer the reader to [338], [386] and [340] for a full introduction.

An MDP is defined as a tuple $\langle S, A, T, R, \gamma \rangle$ where $S \in \{s_1, \ldots, s_n\}$ is a finite set of states, $A \in \{a_1, \ldots, a_m\}$ a finite set of system actions, $T : S \times A \times S \to [0, 1]$ a probabilistic transition function, $R : S \times A \to \mathbb{R}$ a reward function and $\gamma \in [0, 1]$ a factor to discount future rewards. At each

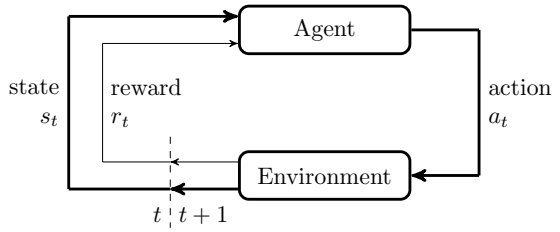


Figure 2.1: The agent-environment interface from [338].

time step t, the system is confronted with some state s_t, performs some action a_t which yields a reward $r_{t+1} : R(s_t, a_t)$ and some state s_{t+1} following the probability distribution $T(s_t, a_t)$. A series of these states, actions and rewards from the onset to some terminal state t_T is called a trajectory $tr : \langle s_{t_0}, a_{t_0}, r_{t_1}, s_{t_1}, \ldots, s_{t_T-1}, a_{t_T-1}, r_{t_T}, s_{t_T} \rangle$. These trajectories typically contain the interaction histories for users with the system. A single trajectory can describe a single session of the user interacting with the system or can contain many different separate sessions. Multiple trajectories may be available in a data set $D \in \{tr_1, \ldots, tr_\ell\}$. The goal is to find a policy π^* out of all $\Pi : S \times A \to [0, 1]$ that maximizes the sum of future rewards at any t, given an end time T:

$$G_t : \sum_{k=t}^{T-1} \gamma^{k-t} r_{k+1} \tag{2.1}$$

If some expectation $\mathbb{E}_\pi$ over the future reward for some policy π can be formulated, a value can be assigned to some state s given that policy:

$$V_\pi(s) = \mathbb{E}_\pi[G_t | s_t = s] \tag{2.2}$$

Similarly, a value can be assigned to an action a in a state s:

$$Q_\pi(s, a) = \mathbb{E}_\pi[G_t | s_t = s, a_t = a] \tag{2.3}$$

Now the optimal policy π^* should satisfy $\forall s \in S, \forall \pi \in \Pi : V_{\pi^*}(s) \geq V_\pi(s)$ and $\forall s \in S, a \in A, \forall \pi \in \Pi : Q_{\pi^*}(s, a) \geq Q_\pi(s, a)$. Assuming a suitable $\mathbb{E}_{\pi^*}[G]$, π^* consists of selecting the action that is expected to yield the highest sum of rewards:

$$\pi^*(s) = \arg\max_a Q_{\pi^*}(s, a), \forall s \in S, a \in A \tag{2.4}$$

With these definitions in place, we now turn to methods of finding π^*. Such methods can be categorized by considering which elements of the MDP are known. Generally, S, A and γ are determined upfront and known. T and R, on the other hand, may or may not be known. If they are both known, the expectation $\mathbb{E}_\pi[G]$ is directly available and a corresponding π^* can be found analytically. In some settings, however, T and R may be unknown and π^* must be found empirically. This can be done by estimating T, R, V, Q and finally π^*

or a combination thereof using data set D. Thus, if we include approximations in Eq. (2.4), we get:

$$\hat{\pi^*}(s)|D = \arg\max_a \hat{Q}_{\hat{\pi^*}}(s,a)|D, \forall s \in S, a \in A \tag{2.5}$$

As D may lack the required trajectories for a reasonable $\mathbb{E}_{\hat{\pi^*}}[G]$ and may even be empty initially, *exploratory* actions can be selected to enrich D. Such actions need not follow $\hat{\pi^*}$ as in Eq. (2.5) but may be selected through some other mechanism such as sampling from the full action set A randomly.

Having introduced RL briefly, we continue by exploring some strategies in applying this framework to the problem of personalizing systems. We return to our earlier example of a video recommendation task and consider a set of n users $U \in \{u_1, \ldots, u_n\}$. A first way to adapt software systems to an individual users' needs is to define a separate environment, corresponding MDP and RL agent for each user. The overall goal becomes to find a set of optimal policies $\{\pi_1^*, \ldots, \pi_n^*\}$ for a set of environments formalized as MDPs $M : \{M_1 : \langle S_1, A_1, T_1, R_1, \gamma_1\rangle, \ldots, M_n : \langle S_n, A_n, T_n, R_n, \gamma_n\rangle\}$. In the case of approximations as in Eq. (2.5), these are made per MDP based on data set D_i with trajectories only involving that environment. In the running example, videos would be recommended to a user based on previous video recommendations and selections of that particular user. The benefit of isolated MDPs is that differences between T_i and T_j or between R_i and R_j for MDPs $M_i \neq M_j$ are handled naturally, e.g. such differences do not make $\mathbb{E}_{\pi_i}[G]$ incorrect. On the other hand, similarities between T_i, T_j and R_i, R_j cannot be used. For example, consider a video recommendation task with $S_{ij} = \{morning, afternoon, night\}$. If two users $u_i \neq u_j$ are both using a video service in the *morning* state, they may both like to watch a breakfast news broadcast whereas in the *night* state they may both prefer a talk show. Learning such patterns for each environment individually may require a substantial number of trajectories and may be infeasible in some settings, such as those where users cannot be identified across trajectories or those where each user is expected to contribute only one trajectory to D_i.

An alternative approach is to use a single agent and MDP with user-specific information in the state space S and learn a single π^* for all users [P5]. In some settings, users can be described using a function that returns a vector representation of the l features that characterize a user $\phi : U \rightarrow \langle \phi_1(U), \ldots, \phi_l(U)\rangle$. Such a vector could for example contain age, favourite genre and viewing history. If two users $u_j \neq u_i$ have both enjoyed the first "Lord of the Rings" movie and viewer u_j has followed up on a recommendation of its sequel by the system then this sequel may be a suitable recommendation for the other viewer u_i as well. Generally, this approach can be valuable when it is unclear which elements of trajectories of users u_j should be used in determining π_i^*. Conceptually, finding π^* now includes determining u_i's preference for actions given a state and determining the relationship between user preferences. This approach should therefore be able to overcome the negative transfer problem described below when enough trajectories are available. The growth in state

space size, on the other hand, may require an exorbitant number of trajectories in D due to the curse of dimensionality [28]. Thus, ϕ is to be carefully designed or dimensionality reduction techniques are to be used in approaches following this strategy. As a closing remark on this approach to personalization, we note that the distinction between task-related and user-specific information is somewhat artificial as S may already contain $\phi(U)$ in many practical settings and we stress that the distinction is made for illustrative purposes here.

A third category of approaches can be considered as a middle ground between learning a single π^* and learning a π_i^* per user. It is motivated by the idea that users and corresponding environments may be similar. If this is the case, then trajectories D_j from some similar environment $M_j \neq M_i$ may prove useful in estimating $\mathbb{E}_{\pi_i}[G]$. One such an approach is based on clustering [91, 133, 221, 341]. Formally, it requires $q \leq n$ groups $G \in \{g_1, \ldots, g_q\}$ and a mapping function $\Phi : M \to G$. In practice, this mapping function is typically defined on the level of users U or the feature representation $\phi(U)$. An RL agent is defined for every g_p and interacts with all environments $M_i, M_j, \Phi(M_i) = \Phi(M_j) = g_p$. Trajectories in D_i and D_j are concatenated or *pooled* to form a single D_p which is used to approximate $\mathbb{E}_{\hat{\pi_p}}[G]$ for all M_i, M_j. A combined D_p may be orders of magnitude bigger than an isolated D_i, which may result in a much better approximation $\mathbb{E}_{\hat{\pi_p}}[G]|D_p$ and a resulting $\hat{\pi}_p^*(s)|D_p$ that yields a higher reward in all environments. For example, users of the video recommendation service may be clustered by age and users in the 'infant' cluster may generally prefer children's movies over history documentaries. A related approach similarly uses trajectories D_j of other environments $M_j \neq M_i$ but still aims to find environment-specific π_i^*. Trajectories in D_j are weighted during estimation of $\mathbb{E}_{\pi_i}[G]$ using some weighting scheme. This can be understood as a generalization of the pooling approach. First, recall that $\Phi : M \to G$ for the pooling approach and note that it can be rewritten to $\Phi : M \times M \to \{0, 1\}$. The weighting scheme, now, is a generalization where $\Phi : M \times M \to \mathbb{R}$. Finding a suitable Φ can be challenging in itself and depends on the availability of user features, trajectories and the task at hand. Typical strategies are to define Φ in terms of similarity of feature representations of users $[\phi(u_i), \phi(u_j)]$ or similarity of D_i, D_j. The two previous approaches work under the assumption that T_i, T_j and R_i, R_j are similar and that Φ is suitable. If either of these assumptions is not met, pooling data may result in a policy that is suboptimal for both M_i and M_j. This phenomenon is typically referred to as the *negative transfer problem* [258].

2.3 Algorithms

In this section we provide an overview of specific RL techniques and algorithms used for personalization. This overview is the result of our systematic literature review as can be seen in Table 2.4. Figure 2.2 contains a diagram of the discussed techniques. We start with a subset of the full RL problem known as k-armed bandits. We bridge the gap towards the full RL setting with contextual

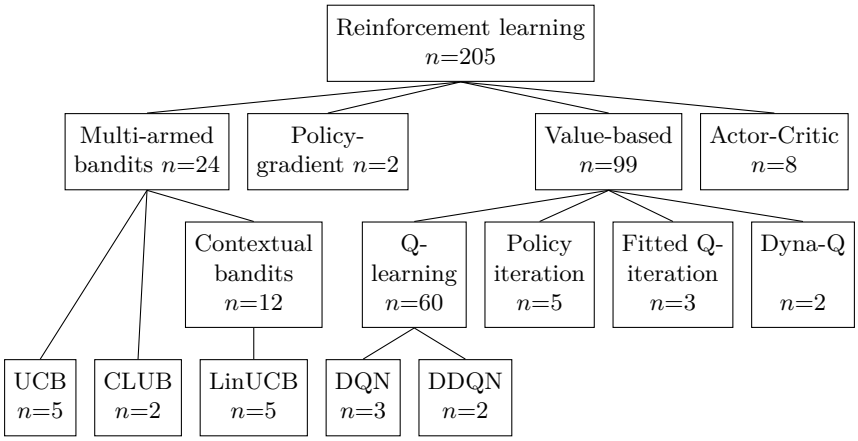


Figure 2.2: Overview of types of RL algorithms discussed in this section and the number of uses in publications included in this survey. See Table 2.4 for a list of all (families of) algorithms used by more than one publication.

bandits approaches. Then, value-based and policy-gradient RL methods are discussed.

2.3.1 Multi-armed bandits

Multi-armed bandits is a simplified setting of RL. As a result, it is often used to introduce basic learning methods that can be extended to full RL algorithms [338]. In the non-associative setting, the objective is to learn how to act optimally in a single situation. Formally, this setting is equivalent to an MDP with a single state. In the associative or *contextual* version of this setting, actions are taken in more than one situation. This setting is closer to the full RL problem yet it lacks an important trait of full RL, namely that the selected action affects the situation. Both associative and non-associative multi-armed bandit approaches do not take into account temporal separation of actions and related rewards.

In general, multi-armed bandit solutions are not suitable when success is achieved by sequences of actions. Non-associative k-armed bandits solutions are only applicable when context is not important. This makes them generally unsuitable for personalizaton as it typically utilizes different personal contexts for different users by offering a different functionality. In some niche areas, however, k-armed bandits are applicable and can be very attractive due to formal guarantees on their performace. If context is of importance, contextual bandit approaches provide a good starting point for personalizing an application. These approaches hold a middle ground between non-associative multi-armed bandits and full RL solutions in terms of modeling power and ease of implementation. Their theoretical guarantees on optimality are less strong than their k-armed counterparts but they are easier to implement, evaluate

and maintain than full RL solutions.

k-Armed bandits

In a k-armed bandit setting, one is constantly faced with the choice between k different actions [338]. Depending on the selected action, a scalar reward is obtained. This reward is drawn from a stationary probability distribution. It is assumed that an independent probability distribution exists for every action. The goal is to maximize the expected total reward over a certain period of time. Still considering the k-armed bandit setting, we assign a value $Q(a)$ to each of the k actions and define this value as the expected reward given that the action was selected. The expected reward given that an action a is selected is defined as follows:

$$Q(a) = \mathbb{E}[r_t | a_t = a]. \tag{2.6}$$

In a trivial problem setting, one knows the exact value of each action and selecting the action with the highest value would constitute the optimal policy. In more realistic problems, it is fair to assume that one cannot know the values of the actions exactly. In this case, one can estimate the value of an action. We denote this estimated value with $\hat{Q}(a)$ and our goal is to have estimate $\hat{Q}(a)$ as close to the true $Q(a)$ as possible.

At each time step t, estimates of the values of actions are obtained. Always selecting the actions with the highest estimated value is called greedy action selection. In this case we are exploiting the knowledge we have built about the values of the actions. When we select actions with a lower expected value, we say we are exploring. In this case we are improving the estimates of values for these actions. In the balancing act of exploration and exploitation, we opt for exploitation to maximize the expected total reward for the next step, while opting for exploration could results in higher expected total reward in the long run.

Action-value methods for multi-armed bandits

Action-value methods [338] denote a collections of methods used for estimating the values of actions. The most natural way of estimating the action-values is to average the rewards that were observed. This method is called the sample-average method. The value estimate $\hat{Q}_\pi(a)$ is then defined as:

$$\hat{Q}(a) = \frac{\sum_{i=1}^{t-1} r_i \cdot \mathbb{1}_{a_i=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{a_i=a}} \tag{2.7}$$

where $\mathbb{1}_{a_i=a}$ is 1 when $a_i = a$ is true and 0 otherwise. A default value is assigned to $\hat{Q}(a)$ when the denominator is zero. As the denominator approaches infinity, the estimate $\hat{Q}(a)$ converges to the true $Q(a)$. Again, the most basic way of selecting actions is the greedy action selection method. Here the action with the highest value is selected. In the case of a tie, one action is selected using tie-breaking methods such as random selection. Greedy action selection is defined

as follows for any time point t:

$$a_t = \arg\max_{a \in A} \hat{Q}(a). \tag{2.8}$$

Greedy action selection only exploits knowledge built up using the action-value method and only maximizes the immediate reward. This can lead to incorrect action-value approximations because actions with e.g. low *estimated* but high *actual* values are not sampled. An improvement over this greedy action selection is to randomly explore with a small probability ϵ. This method is named the ϵ-greedy action selection. A benefit of this method is that, while it is relatively simple, in the limit $\hat{Q}(a)$ will converge to $Q(a)$ [338]. This indicates that the probability of selecting the optimal action is then greater than $1 - \epsilon$ which is near certainty.

Incremental Implementation

In Section 2.3.1 we discussed a method to estimate action-values using sample-averaging. To ensure the usability of these method in real-world applications, we need to be able to compute these values in an efficient way. Assume a setting with one action. At each iteration j a reward r_{t_j} is obtained after selecting an action. Let $\hat{Q}_n(a)$ denote the estimate value of the action after $n-1$ iterations. We can then define:

$$\hat{Q}_n(a) = \frac{r_{t_1} + r_{t_2} + r_{t_3} + ... + r_{t_{n-1}}}{n-1}. \tag{2.9}$$

Using this approach would mean storing the values of all the rewards to recalculate $\hat{Q}_n(a)$ from scratch at every iteration. There is however a more efficient way for calculating $\hat{Q}_n(a)$ that is constant in memory and computation time. Rewriting it yields the following update rule:

$$\hat{Q}_{n+1}(a) = \hat{Q}_n(a) + \frac{1}{n}[r_{t_n} - \hat{Q}_n(a)], \tag{2.10}$$

where the term $\hat{Q}_n(a)$ represents the old estimate, $[r_n - \hat{Q}_n(a)]$ the error in the estimate we made of the reward and $\frac{1}{n}$ the learning rate.

UCB: Upper-Confidence Bound

The greedy and ϵ-greedy action selection methods were discussed in Section 2.3.1 and it was introduced that exploration is required to establish good action-value estimates. Although ϵ-greedy explores all actions eventually, it does so randomly. A better way of exploration would take into account the action-value's proximity to the optimal value and the uncertainty in the value estimations. Intuitively, we want a selected action a to either provide a good immediate reward or else some very useful information in updating $\hat{Q}(a)$. An

approach that uses this idea is the upper confidence bound action selection (UCB) method [15, 115, 338]. UCB is defined as follows at time step t:

$$a_t = \arg\max_a \left[\hat{Q}_n(a) + c \cdot \sqrt{\frac{\ln t}{N_t(a)}} \right] \tag{2.11}$$

where $N_t(a)$ is how often action a was chosen up to time t and $c > 0$ is a parameter to control the rate of exploration. The square root term denotes the level of uncertainty in the approximation of the value of action a. Hence, UCB provides an upper bound for the true value of the action a. Here, c is used to define the confidence level. When the action a is selected often, $N_t(a)$ will become larger which leads the uncertainty term to decrease. On the other hand, if the action a is not selected very often, t increases and so does the uncertainty term.

k-Armed bandit approaches address the trade-off between exploitation and exploration directly. It has been shown that the difference between the obtained rewards and optimal rewards, or the *regret*, is at best logarithmic in the number of iterations n in the absence of prior knowledge of the action value distributions and in the absence of context [187]. UCB algorithms with a regret logarithmic in and uniformly distributed over n exist [15]. This makes them a very interesting choice when strong theoretical guarantees on performance are required.

Whether these algorithms are suitable, however, depends on the setting at hand. If there is a large number of actions to choose from or when the task is not stationary k-armed bandits are typically too simplistic. In a news recommendation task, for example, exploration may take longer than an item stays relevant. Additionally, k-armed bandits are not suitable when action values are conditioned on the situation at hand, that is: when a single action results in a different reward based on e.g. time-of-day or user-specific information such as in Section 2.2. In these scenarios, the problem formalization of contextual bandits and the use of function approximation are of interest.

Contextual bandits

In the previous sections, action-values where not associated with different situations. In this section we extend the non-associative bandit setting to the associative setting of contextual bandits. Assume a setting with n k-armed bandits problems. At each time step t one encounters a situation with a randomly selected k-armed bandits problem. We can use some of the approaches that were discussed to estimate the action values. However, this is only possible if the true action-values change slowly between the different n problems [338]. Add to this setting the fact that now at each time t a distinctive piece of information is provided about the underlying k-armed bandit which is not the actual action value. Using this information we can now learn a policy that uses the distinctive information to associate the k-armed bandit with the best action to take. This approach is called contextual bandits and uses trial-and-error to

search for the optimal actions and associates these actions with situation in which they perform optimally. This type of algorithm is positioned between k-armed bandits and full RL. The similarity with RL lies in the fact that a policy is learned while the association with k-armed bandits stems from the fact that actions only affect immediate rewards. When actions are allowed to affect the next situation as well then we are dealing with RL.

Function approximation: LinUCB and CLUB

Despite the good theoretical characteristics of the UCB algorithm, it is not often used in the contextual setting in practice. The reason is that in practice, state and action spaces may be very large and although UCB is optimal in the uninformed case, we may do better if we use obtained information across actions and situations. Instead of maintaining isolated sample-average estimates per action or per state-action pair such as in Sections 2.3.1 and 2.3.1, we can estimate a parametric payoff function approximated from data. The parametric function takes some feature description of actions for k-armed bandit settings and state-action pairs for the contextual bandit setting and output some estimated $\hat{Q_\theta}(a)$. Here, we focus on the contextual-bandit algorithms LinUCB and CLUB.

LinUCB (Linear Upper-Confidence Bound) uses linear function approximation to calculate the confidence interval efficiently in closed form [197]. Define the expected payoff for action a with the d-dimensional featurized state $s_{t,a}$ and Θ_a^* a vector of unknown parameters as follows:

$$\mathbb{E}[r_a|s_a] = s_a^T \Theta_a^*. \tag{2.12}$$

Using ridge regression, an estimate of $\hat{\Theta}_a$ can be obtained [197]. Consequently, it can be shows that for any $\sigma > 0$ and $s_a \in \mathbb{R}^d$ with $\alpha = 1 + \sqrt{ln(\frac{2}{\sigma})/2}$ a reasonably tight estimate for the expected payoff of arm a can be obtained as follows:

$$a_t = \arg\max_a \left[s_a^T \Theta_a^* + \alpha \sqrt{s_a^T A_a^{-1} s_a} \right], \tag{2.13}$$

where $A_a^{-1} = D_a^T D_a + I_d$ and D_a a design matrix of dimension m x d whose rows are the m contexts that are observed, $b_a \in \mathbb{R}^m$ the corresponding response vector and I_d the d x d identity matrix [197].

Similar to LinUCB, CLUB (Clustering of bandits) utilizes the linear bandit algorithm for payoff estimation [121]. In contrast to LinUCB, CLUB uses adaptive clustering in order to speed up the learning process. The main idea is to use confidence balls of user models estimate user similarity and share feedback across similar users. CLUB can thus be understood as a cluster-based alternative (see Section 2.2) to LinUCB algorithm.

2.3.2 Value-based RL

In value based RL, we learn an estimate V of the optimal value function V_{π^*} for a given policy π. We do this with the aim of finding π^*. Temporal-difference

Algorithm 1 Sarsa - An on-policy temporal-difference RL algorithm

Parameters: learning rate $\alpha \in (0, 1]$ and $\epsilon > 0.0$.

Initialize $\hat{Q}_\pi \; \forall \; s \in S$, $a \in A$. For terminal states initialize the value with 0.
for all episodes **do**
  Initialize s
  Choose action a in s using π derived from $\hat{Q}_\pi$ (e.g. ϵ-greedy)
  for all steps in episode **do**
    Select action a and obtain reward r and next state s'
    Take next action a' from s' following π derived from $\hat{Q}_\pi$ (e.g. ϵ-greedy)
    $\hat{Q}_\pi(s, a) = \hat{Q}_\pi(s, a) + \alpha \left[r + \gamma \hat{Q}_\pi(s', a') - \hat{Q}_\pi(s, a) \right]$
    Set $s = s'$ and $a = a'$
    Stop loop if s is terminal
  end for
end for

(TD) prediction is a method that learns from raw experiences without having to build a model of the environment the policy is interacting with [338]. In this section, we discuss various RL algorithms based on TD prediction.

Sarsa: on-policy temporal-difference RL

Sarsa is an on-policy temporal-difference method that learns an action-value function [328, 338]. Given the current behaviour policy π, we estimate $\hat{Q_\pi}(a)$ $\forall$ s, and a. This is done using transitions from state-action pair to state-action pair. Events of the form $\langle s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1} \rangle$ are used in the following update rule to estimate the state-action values:

$$\hat{Q}_\pi(s_t, a_t) = \hat{Q}_\pi(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \hat{Q}_\pi(s_{t+1}, a_{t+1}) - \hat{Q}_\pi(s_t, a_t) \right]. \quad (2.14)$$

This update rule is applied after every transition from s_t to s_{t+1}. In case s_{t+1} is a terminal state, a value of zero is assigned. By doing this we are ensuring that the estimate $\hat{Q_\pi}$ for a behaviour policy π while resulting in changes in π given Q_π. Sarsa will converge to an optimal action-value function Q_{π^*} and hence an optimal policy π^* in the limit given that all possible state-action pairs are visited an infinite amount of time [338]. Consequently, Sarsa converges to the greedy policy in the limit. Algorithm 1 shows Sarsa in more detail.

Q-learning: off-policy temporal-difference RL

Q-learning was one of the breakthroughs in the field of RL [338, 384]. Q-learning is classified as an off-policy temporal-difference algorithm for control. Similar to Sarsa, Q-learning approximates the optimal action-value function Q_{π^*} by learning $\hat{Q_{\pi^*}}$. Differently from Sarsa, Q-learning learns $\hat{Q_{\pi^*}}$ independently of the policy being followed. The policy being followed still has an effect

Algorithm 2 Q-Learning - An off-policy Temporal-Difference RL algorithm

Parameters: learning rate $\alpha \in (0, 1]$ and $\epsilon > 0$.

Initialize $\hat{Q}_\pi \; \forall \; s \in S$, $a \in A$. For terminal states initialize the value with 0.
for all episodes **do**
  Initialize s
  for all steps in episode **do**
    Choose action a in s using π derived from $\hat{Q}_\pi$ (e.g. ϵ-greedy)
    Take action a and obtain reward r and next state s'
    $\hat{Q}_\pi(s,a) = \hat{Q}_\pi(s,a) + \alpha \left[r + \gamma \cdot \arg\max_a \hat{Q}_\pi(s',a) - \hat{Q}_\pi(s,a) \right]$
    Set $s = s'$
    Stop loop if s is terminal
  end for
end for

on the learning process, but only by determining which state-action pairs are visited and consequently updated. Algorithm 2 shows Q-learning in more detail. The update rule for Q-learning is defined as follows:

$$\hat{Q}_\pi(s_t, a_t) = \hat{Q}_\pi(s_t, a_t) + \alpha \left[r_{t+1} + \gamma max_a \hat{Q}_\pi(s_{t+1}, a) - \hat{Q}_\pi(s_t, a_t) \right]. \quad (2.15)$$

Value-function approximation

In sections 2.3.2 and 2.3.2 we discussed tabular algorithms for value-based RL. In this section we discuss function approximation in RL for estimating state-value functions from a known policy π (i.e. on-policy RL). The difference with the tabular approach is that we represent v_π as a parameterized function with a weight vector $w \in \mathbb{R}^d$ where $\hat{v}(s, w) \approx v_\pi(s)$ is the approximated value of state s given the learned weights w. Different function approximators can be used to estimate $\hat{v}$. For instance, $\hat{v}$ can be a deep neural network with w representing the weights of the network. In the tabular version of value-based RL, states and their estimated values are isolated from each other while in function approximation adjusting one weight in the network can lead to changes in the estimated values of many states. This form of learning is powerful due its ability to generalize across different states, but at the same time may lead to more complex models that are harder to understand and to tune. An example of value-function approximation is the deep Q-network (DQN) algorithm [239]. This algorithm combines deep (convolutional) neural network and Q-learning. Using DQN, it was shown that RL agents can achieve state-of-the-art performances on many problems without relying on engineered features. DNQ learns directly from raw (pixel) data instead. The following update rule is an alteration of the Q-learning (semi-gradient of Q-learning [338]) update rule for

estimating the weights of the network:

$$w_{t+1} = w_t + \alpha \left[r_{t+1} + \gamma \cdot \max_a \hat{Q}_\pi(s_{t+1}, a, w_t) - \hat{Q}_\pi(s_t, a_t, w_t) \right] \nabla_{wt} \hat{Q}_\pi(s_t, a_t, w_t). \tag{2.16}$$

2.3.3 Policy-gradient RL

In value-based RL values of actions are approximated and then a policy is derived by selecting actions using a certain selection strategy. In policy-gradient RL we learn a parameterized policy directly [338, 339]. Consequently, we can select actions without the need for an explicit value function. Let $\Theta \in \mathbb{R}^d$ where d is the dimension of the parameter vector Θ. For policy-based methods that also rely on a value function, we denote the function's weight vector denoted by $w \in \mathbb{R}^{d'}$ as $\hat{v}(s, w)$. Define the probability of selecting action a at time step t given state s with policy parameters Θ as:

$$\pi(a|s, \Theta) = P[a_t = a | s_t = s, \Theta_t = \Theta] \tag{2.17}$$

Consider a function $J(\Theta)$ that quantifies the performance of the policy π with respect to parameter vector Θ. The goal is to optimize Θ such that $J(\Theta)$ is maximized. We use the following update rule to approximate gradient ascent in J where the term $\widehat{\nabla J(\Theta_t)} \in \mathbb{R}^d$ approximates the gradient of $J(\Theta)$ at t:

$$\Theta_{t+1} = \Theta_t + \alpha \widehat{\nabla J(\Theta_t)}. \tag{2.18}$$

2.3.4 Actor-critic

In actor-critic methods [182, 338] both the value and policy functions are approximated. The actor in actor-critic is the learned policy while the critic approximates the value function. Algorithm 3 shows the one-step episodic actor-critic algorithm in more detail. The update rule for the parameter vector Θ is defined as follows:

$$\Theta_{t+1} = \Theta_t + \alpha \delta_t \frac{\nabla \pi(a|s_t, \Theta_t)}{\pi(a|s_t, \Theta_t)} \tag{2.19}$$

where δ_t is defined as follows:

$$\delta_t = r_{t+1} + \gamma \hat{v}(s_{t+1}, w) - \hat{v}(s_t, w). \tag{2.20}$$

2.4 A classification of personalization settings

Personalization has many different definitions [55, 93, 294]. We adopt the definition proposed in [93] as it is based on 21 existing definitions found in literature and suits a variety of application domains: "personalization is a process that

Algorithm 3 One-step episodic actor-critic

Input: differentiable policy $\pi(a|s, \Theta)$ and state-value function $\hat{v}(s, w)$
Parameters: $\alpha(\Theta) > 0$ and $\alpha(w) > 0$

Initialize $\Theta \in \mathbb{R}^d$ and $w \in \mathbb{R}^{d'}$
for all episodes **do**
  Initialize S
  I = 1
  for all step in episode **do**
    Choose action a in s using π: $a \quad \pi(.|s, \Theta)$
    Take action a and obtain reward r and next state s'
    $\delta = r + \gamma\hat{v}(s', w) - \hat{v}(s, w)$
    $w = w + \alpha(w)\delta\nabla\hat{v}(s, w)$
    $\Theta = \Theta + \alpha(\Theta)I\delta\nabla \ln \pi(a|s, \Theta)$
    $I = \gamma I$
    $s = s'$
  end for
end for

changes the functionality, interface, information access and content, or distinctiveness of a system to increase its personal relevance to an individual or a category of individuals". This definition identifies personalization as a process and mentions an existing system subject to that process. We include aspects of both the desired process of change and existing system in our framework. Section 2.5.4 further details how this framework was used in a SLR.

Table 2.1 provides an overview of the framework. On a high level, we distinguish three categories. The first category contains aspects of suitability of system behavior. We differentiate settings in which suitability of system behavior is determined explicitly by users and settings in which it is inferred by the system after observing user behavior [309]. For example, a user can explicitly rate suitability of a video recommendation; a system can also infer suitability by observing whether the user decides to watch the video. Whether implicit or explicit feedback is preferable depends on availability and quality of feedback signals [163, 309]. Besides suitability, we consider safety of system behavior. Unaltered RL algorithms use trial-and-error style exploration to optimize their behavior yet this may not suit a particular domain. For example, tailoring the insulin delivery policy of an artificial pancreas to the metabolism of an individual requires trial insulin delivery action but these should only be sampled when their outcome is within safe certainty bounds [76]. If safety is a significant concern in the systems' application domain, specifically designed safety-aware RL techniques may be required, see [264] and [114] for overviews of such techniques.

Aspects in the second category deal with the availability of upfront knowledge. Firstly, knowledge of how users respond to system actions may be captured in user models. Such models open up a range of RL solutions that require

Table 2.1: Framework to categorize personalization setting by.

Category	A#	Aspect	Description	Range
Suitability outcome	A1	Control	The extent to which the user defines the suitability of behavior explicitly.	Explicit - implicit
	A2	Safety	The extent to which safety is of importance.	Trivial - critical
Upfront knowledge	A3	User models	The a priori availability of models that describe user responses to system behavior.	Unavailable - unlimited
	A4	Data availability	The a priori availability of human responses to system behavior.	Unavailable - unlimited
New Experiences	A5	Interaction availability	The availability of new samples of interactions with individuals.	Unavailable - unlimited
	A6	Privacy sensitivity	The degree to which privacy is a concern.	Trivial - critical
	A7	State observability	The degree to which all information to base personalization can be measured.	Partial - full

less or no sampling of new interactions with users [149]. As an example, user pain models are used to predict suitability of exercises in an adaptive physical rehabilitation curriculum manager a priori[363]. Models can also be used to interact with the RL agent in simulation. For example, dialogue agent modules may be trained by interacting with a simulated chatbot user as in Chapter 3. Secondly, upfront knowledge may be available in the form of data on human responses to system behavior. This data can be used to derive user models and can be used to optimize policies directly and provide high-confidence evaluations of such policies [203, 355].

The third category details new experiences. Empirical RL approaches have proven capable of modelling extremely complex dynamics, however, this typically requires complex estimators that in turn need substantial amounts of training data. The availability of users to interact with is therefore a major consideration when designing an RL solution. A second aspect that relates to the use of new experiences is privacy sensitivity of the setting. Privacy sensitivity is of importance as it may restrict sharing, pooling or any other specific usage of data [17]. Finally, we identify the state observability as a relevant aspect. In some settings, the true environment state cannot be observed directly but must be estimated using available observations. This may be common as personalization exploits differences in mental [40, 177, 381] and physical state [118, 222]. For example, recommending appropriate music during running involves matching songs to the user emotional state and e.g. running pace. Both mental and physical state may be hard to measure accurately [2, 33, 271].

Although aspects in Table 2.1 are presented separately, we explicitly note that they are not mutually independent. Settings where privacy is a major concern, for example, are expected to typically have less existing and new interactions available. Similarly, safety requirements will impact new interaction availability. Presence of upfront knowledge is mostly of interest in settings where control lies with the system as it may ease the control task. In contrast, user models may be marginally important if desired behavior is specified by the user in full. Finally, a lack of upfront knowledge and partial observability complicates adhering to safety requirements.

2.5 A systematic literature review

A SLR is 'a form of secondary study that uses a well-defined methodology to identify, analyze and interpret all available evidence related to a specific research question in a way that is unbiased and (to a degree) repeatable' [41]. PRISMA is a standard for reporting on SLRs and details eligibility criteria, article collection, screening process, data extraction and data synthesis [243]. This section contains a report on this SLR according to the PRISMA statement. This SLR was a collaborative work to which all authors contributed. We denote authors by abbreviation of their names, e.g. FDH, EG, AEH and MH.

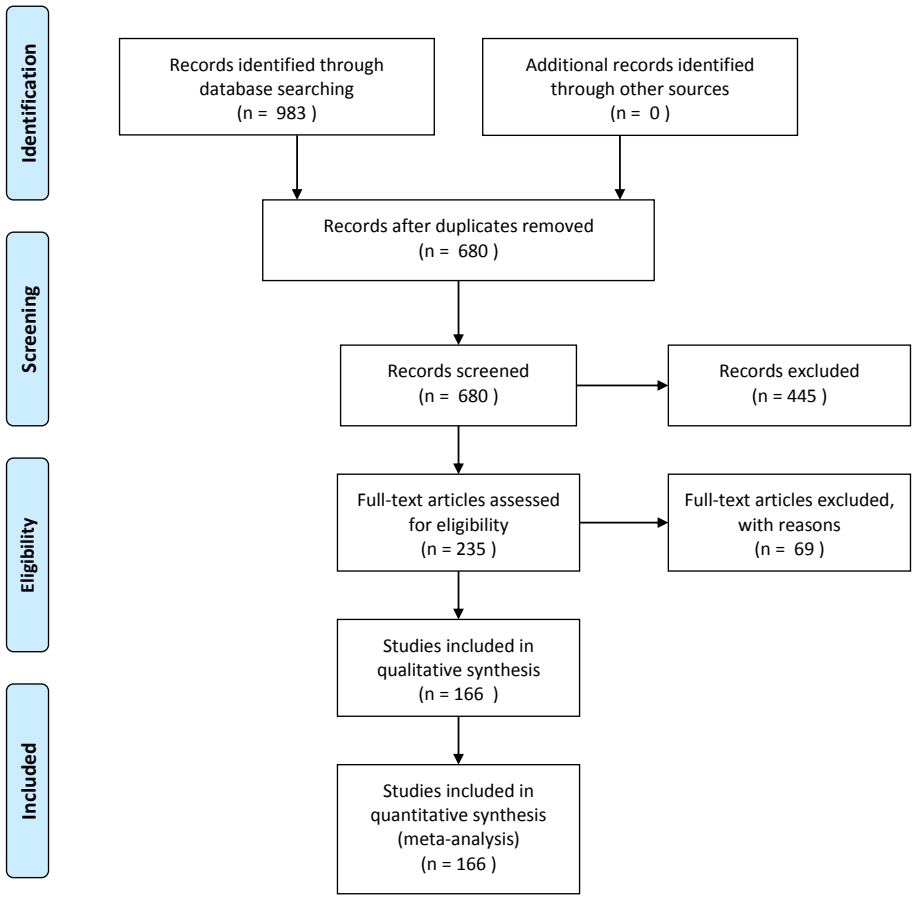


Figure 2.3: Overview of the SLR process.

2.5.1 Inclusion criteria

Studies in this SLR were included on the basis of three eligibility criteria. To be included, articles had to be published in a peer-reviewed journal or conference proceedings in English. Secondly, the study had to address a problem fitting to our definition of personalization as described in Section 2.4. Finally, the study had to use a RL algorithm to address such a personalization problem. Here, we view contextual bandit algorithms as a subset of RL algorithms and thus included them in our analysis. Additionally, we excluded studies in which a RL algorithm was used for purposes other than personalization.

2.5.2 Search strategy

Figure 2.3 contains an overview of the SLR process. The first step is to run a query on a set of databases. For this SLR, a query was run on Scopus, IEEE Xplore, ACM's full-text collection, DBLP and Google Scholar on June 6, 2018. These databases were selected as their combined index spans a wide range, and their combined result set was sufficiently large for this study. Scopus and IEEE Xplore support queries on title, keywords and abstract. ACM's full-text collection, DBLP and Google scholar do not support queries on keywords and abstract content. We therefore ran two kinds of queries: we queried on title only for ACM's full-text collection, DBLP and Google Scholar and we extended this query to keywords and abstract content for Scopus and IEEE Xplore. The query was constructed by combining techniques of interest and keywords for the personalization problem. For techniques of interest the terms 'reinforcement learning' and 'contextual bandits' were used. For the personalization problem, variations on the words 'personalized', 'customized', 'individualized' and 'tailored' were included in British and American spelling. All queries are listed in Appendix A. Query results were de-duplicated and stored in a spreadsheet.

2.5.3 Screening process

In the screening process, all query results are tested against the inclusion criteria from Section 2.5.1 in two phases. We used all criteria in both phases. In the first phase, we assessed eligibility based on keywords, abstract and title whereas we used full text of the article in the second phase. In the first phase, a spreadsheet with de-duplicated results was shared with all authors via Google Drive. Studies were assigned randomly to authors who scored each study by the eligibility criteria. The results of this screening were verified by one of the other authors, assigned randomly. Disagreements were settled in meetings involving those in disagreement and FDH if necessary. In addition to eligibility results, author preferences for full-text screening were recorded on a three-point scale. Studies that were not considered eligible were not taken into account beyond this point, all other studies were included in the second phase.

In the second phase, data on eligible studies was copied to a new spreadsheet. This sheet was again shared via Google Drive. Full texts were retrieved and evenly divided amongst authors according to preference. For each study, the assigned author then assessed eligibility based on full text and extracted the data items detailed below.

2.5.4 Data items

Data on setting, solution and methodology were collected. Table 2.2 contains all data items for this SLR. For data on setting, we operationalized our framework from Table 2.1 in Section 2.4. To assess trends in solution, algorithms used, number of MDP models (see Section 2.2) and training regime were recorded. Specifically, we noted whether training was performed by interacting

with actual users ('live'), using existing data and a simulator of user behavior. For the algorithms, we recorded the name as used by the authors. To gauge maturity of the proposed solutions and the field as a whole, data on the evaluation strategy and baselines used were extracted. Again, we listed whether evaluation included 'live' interaction with users, existing interactions between systems and users or using a simulator. Finally, publication year and application domain were registered to enable identification of trends over time and across domains. The list of domains was composed as follows: during phase one of the screening process, all authors recorded a domain for each included paper, yielding a highly inconsistent initial set of domains. This set was simplified into a more consistent set of domains which was used during full-text screening. For papers that did not fall into this consistent set of domains, two categories were added: a 'Domain Independent' and an 'Other' category. The actual domain was recorded for the five papers in the 'Other' category. These domains were not further consolidated as all five papers were assigned to unique domains not encountered before.

2.5.5 Synthesis and analysis

To facilitate analysis, reported algorithms were normalized using simple text normalization and key-collision methods. The resulting mappings are available in the dataset release [80]. Data was summarized using descriptive statistics and figures with an accompanying narrative to gain insight into trends with respect to settings, solutions and evaluation over time and across domains.

2.6 Results

The quantitative synthesis and analyses introduced in Section 2.5.5 were applied to the collected data. In this section, we present insights obtained. We focus on the major insights and encourage the reader to explore the tabular view in Appendix A.1 or the collected data for further analysis [80].

Before diving into the details of the study in light of the classification scheme we have proposed, let us first study some general trends. Figure 2.4 shows the number of publications addressing personalization using RL techniques over time. A clear increase can be seen. With over forty entries, the health domain contains by far the most articles, followed by entertainment, education and commerce with all approximately just over twenty five entries. Other domains contain less than twelve papers in total. Figure 2.5a shows the popularity of domains for the five most recent years and seems to indicate that the number of articles in the health domain is steadily growing, in contrast with the other domains. Of course, these graphs are based on a limited number of publications, so drawing strong conclusions from these results is difficult. We do need to take into account that the popularity of RL for personalization is increasing in general. Therefore Figure 2.5b shows the relative distribution of studies over domains for the five most recent years. Now we see that the health domain

Table 2.2: Data items in SLR. The last column relates data items to aspects of setting from Table 2.1 where applicable.

	#	Data item	Values	A#
Setting	1	User defines suitability of system behavior explicitly	Yes, No	A1
	2	Suitability of system behavior is derived	Yes, No	A1
	3	Safety is mentioned as a concern in the article	Yes, No	A2
	4	Privacy is mentioned as a concern in the article	Yes, No	A6
	5	Models of user responses to system behavior are available	Yes, No	A3
	6	Data on user responses to system behavior are available	Yes. No	A4
	7	New interactions with users can be sampled with ease	Yes, No	A5
	8	All information to base personalization on can be measured	Yes, No	A7
Solution	9	Algorithms	N/A	–
	10	Number of learners	1, 1/user, 1/group, multiple	–
	11	Usage of traits of the user	state, other, not used	–
	12	Training mode	online, batch, other, unknown	–
	13	Training in simulation	Yes, No	A3
	14	Training on a real-life dataset	Yes, No	A4
	15	Training in 'live' setting	Yes, No	A5
Evaluation	16	Evaluation in simulation	Yes, No	A3
	17	Evaluation on a real-life dataset	Yes, No	A4
	18	Evaluation in 'live' setting	Yes, No	A5
	19	Comparison with 'no personalization'	Yes, No	–
	20	Comparison with non-RL methods	Yes, No	–

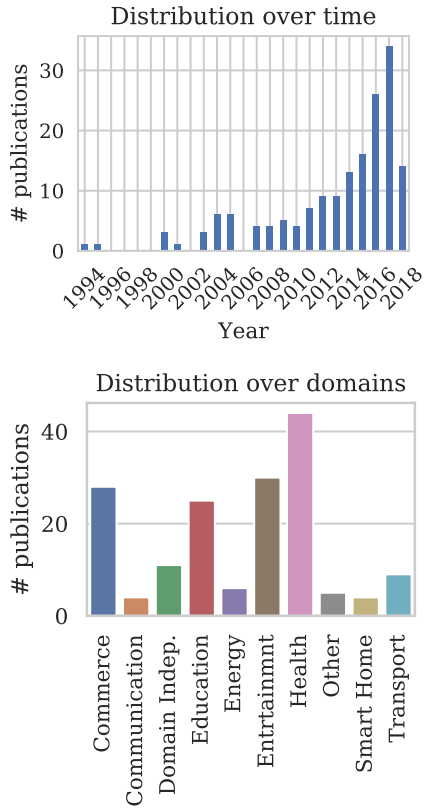


Figure 2.4: Distribution of included papers over time and over domains. Note that only studies published prior to the query date of June 6, 2018 were included.

is just following the overall trend, and is not becoming more popular within studies that use RL for personalization. We fail to identify clear trends for other domains from these figures.

2.6.1 Setting

Table 2.3 provides an overview of the data related to setting in which the studies were conducted. The table shows that user responses to system behavior are present in a minority of cases (66/166). Additionally, models of user behavior are only used in around one quarter of all publications. The suitability of system behavior is much more frequently derived from data (130/166) rather than explicitly collected by users (39/166). Privacy is clearly not within the scope of most articles, only in 9 out of 166 cases do we see this issue explicitly mentioned. Safety concerns, however, are mentioned in a reasonable proportion of studies (30/166). Interactions can generally be sampled with ease and the resulting information is frequently sufficient to base personalization of the system at

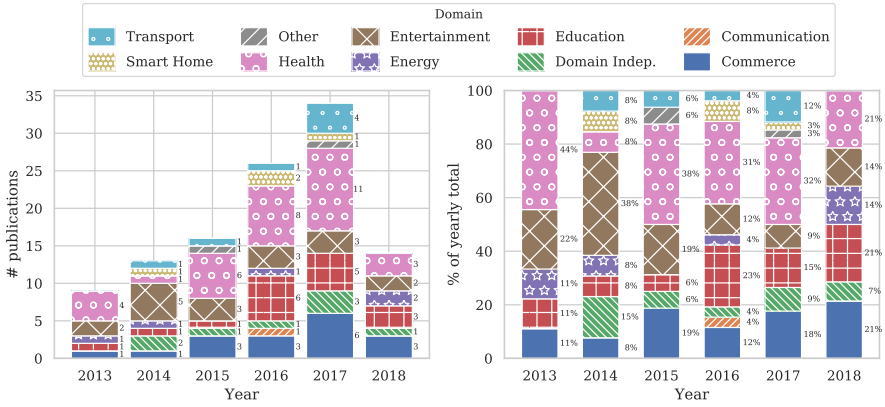


Figure 2.5: Popularity of domains for the five most recent years.

Table 2.3: Number of publications by aspects of setting.

Aspect	#
User defines suitability of system behavior explicitly	39
Suitability of system behavior is derived	130
Safety is mentioned as a concern in the article	30
Privacy is mentioned as a concern in the article	9
Models of user responses to system behavior are available	41
Data on user responses to system behavior are available	66
New interactions with users can be sampled with ease	97
All information to base personalization on can be measured	132

hand on.

Let us dive into some aspects in a bit more detail. A first trend we anticipate is an increase of the fraction of studies working with real data on human responses over the years, considering the digitization trend and associated data collection. Figure 2.6a shows the fraction of papers for which data on user responses to system behavior is available over time. Surprisingly, we see that this fraction does not show any clear trend over time. Another aspect of interest relates to safety issues in particular domains. We hypothesize that in certain domains, such as health, safety is more frequently mentioned as a concern. Figure 2.6b shows the fraction of papers of the different domains in which safety is mentioned. Indeed, we clearly see that certain domains mention safety much more frequently than other domains. Third, we explore the ease with which interactions with users can be sampled. Again, we expect to see substantial differences between domains. Figure 2.7 confirms our intuition. Interactions can be sampled with ease more frequently in studies in the commerce, entertainment, energy, and smart homes domains when compared to communication and health domains.

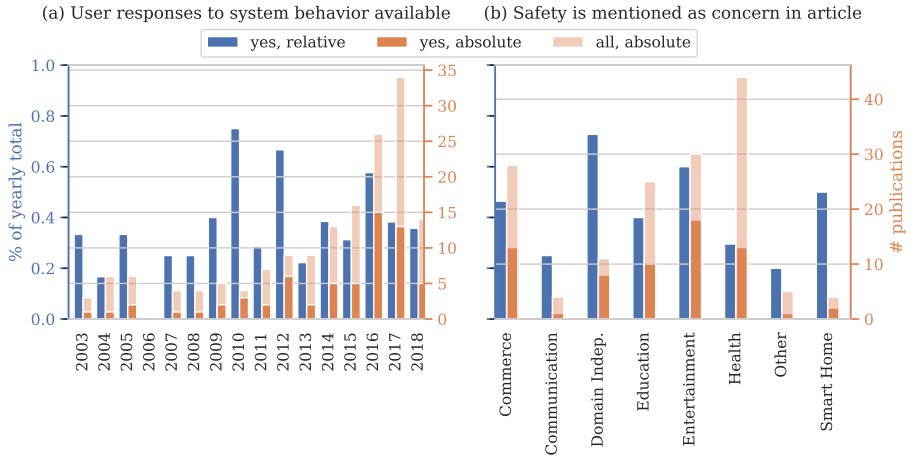


Figure 2.6: Availability of user responses over time (a), and mentions of safety as a concern over domains (b).

Finally, we investigate whether upfront knowledge is available. In our analysis, we explore both real data as as well user models being available upfront. One would expect papers to have at least one of these two prior to starting experiments. User models and not real data were reported in 41 studies, while 53 articles used real data but no user model and 12 use both. We see that for 71 studies neither is available. In roughly half of these, simulators were used for both training (38/71) and evaluation (37/71). In a minority, training (15/71) and evaluation (17/71) were performed in a live setting, e.g. while collecting data.

2.6.2 Solution

In our investigation into solutions, we first explore the algorithms that were used. Figure 2.8 shows the distribution of usage frequency. A vast majority of the algorithms are used only once, some techniques are used a couple of times and one algorithm is used 60 times. Note again that we use the name of the algorithms used by the authors as a basis for this analysis. Table 2.4 lists the algorithms that were used more than once. A significant number of studies (60/166) use the Q-learning algorithm. At the same time, a substantial number of articles (18/166) reports the use of RL as the underlying algorithmic framework without specifying an actual algorithm. The contextual bandits, Sarsa, actor-critic and inverse RL (IRL) algorithms are used in respectively (18/166), (12/166), (8/166), (8/166) and (7/166) papers. We also observe some additional algorithms from the contextual bandits family, such as UCB and LinUCB. Furthermore, we find various mentions that indicate the usage of deep neural networks: deep reinforcement learning, DQN and DDQN. In general, we find that some publications refer to a specific algorithm whereas

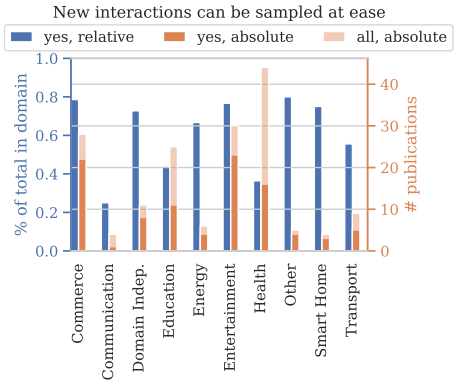


Figure 2.7: New interactions with users can be sampled with ease.

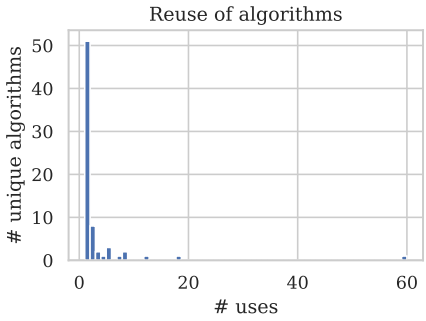


Figure 2.8: Distribution of algorithm usage frequencies.

Table 2.4: Algorithm usage for all algorithms that were used in more than one publication.

Algorithm	# of uses
Q-learning [384]	60
RL, not further specified	18
Contextual bandits	12
Sarsa [336]	8
Actor-critic	8
Inverse reinforcement learning	7
UCB [15]	5
Policy iteration	5
LinUCB [63]	5
Deep reinforcement learning	4
Fitted Q-iteration [295]	3
DQN [239]	3
Interactive reinforcement learning	2
TD-learning	2
DYNA-Q [337]	2
Policy gradient	2
CLUB [121]	2
Monte carlo	2
Thompson sampling	2
DDQN [371]	2

others only report generic techniques or families thereof.

Figure 2.9a lists the number of models used in the included publications. The majority of solutions relies on a single-model architecture. On the other end of the spectrum lies the architecture of using one model per person. This

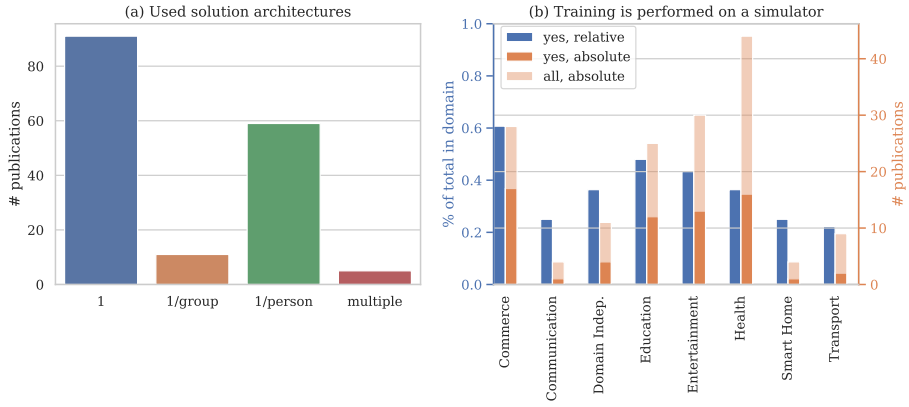


Figure 2.9: Occurence of different solution architectures (a) and usage of simulators in training (b). For (a), publications that compare architectures are represented in the 'multiple' category.

architecture comes second in usage frequency. The architecture that uses one model per group can be considered a middle ground between these former two. In this architecture, only experiences with relevant individuals can be shared. Comparisons between architectures are rare. We continue by investigating whether and where traits of the individual were used in relation to these architectures. Table 2.5 provides an overview. Out of all papers that use one model, 52.7% did not use the traits of the individuals and 41.7 % included traits in the state space. 47.5% of the papers include the traits of the individuals in the state representation while in 37.3% of the papers the traits were not included. In 15.3% of the cases this was not known.

Figure 2.9b shows the popularity of using a simulator for training per domain. We see that a substantial percentage of publications use a simulator and that simulators are used in all domains. Simulators are used in the majority of publications for the energy, transport, communication and entertainment domains. In publications in the first three out of these domains, we typically find applications that require large-scale implementation and have a big impact on infrastructure, e.g. control of the entire energy grid or a fleet of taxis in a large city. This complicates the collection of useful realistic dataset and training in a live setting. This is not the case for the entertainment domain with 17 works using a simulator for training. Further investigation shows that nine out of these 17 also include training on real data or in a 'live' setting. It seems that training on a simulator is part of the validation of the algorithm rather than the prime contribution of the paper in the entertainment domain.

2.6.3 Evaluation

In investigating evaluation rigor, we first turn to the data on which evaluations are based. Figure 2.10 shows how many studies include an evaluation in a

Table 2.5: Number of models and the inclusion of user traits.

Traits of users were used	Number of models			
	1	1/group	1/person	multiple
In state representation	38	8	28	2
Other	5	0	9	3
Not used	48	3	22	0
Total	91	11	59	5

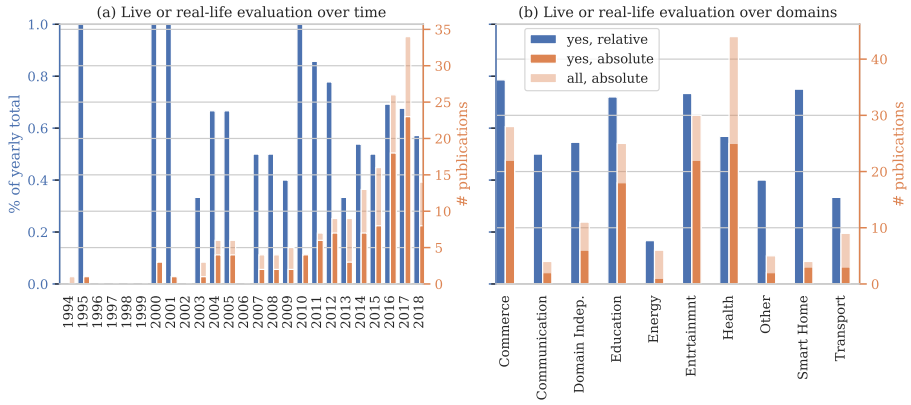


Figure 2.10: Number of papers with a 'live' evaluation or evaluation using data on user responses to system behavior.

'live' setting or using existing interactions with users. In the years up to 2007 few studies were done and most of these included realistic evaluations. In more recent years, the absolute number of studies shows a marked upward trend to which the relative number of articles that include a realistic evaluation fails to keep pace. Figure 2.10 also shows the number of realistic evaluations per domain. Disregarding the smart home domain, as it contains only four studies, the highest ratio of real evaluations can be found in the commerce and entertainment domains, followed by the health domain.

We look at possible reasons for a lack of realistic evaluation using our categorization of settings from Section 2.4. Indeed, there are 63 studies with no realistic evaluation versus 104 with a realistic evaluation. Because these group sizes differ, we include ratios with respect to these totals in Table 2.6. The biggest difference between ratios of studies with and without a realistic evaluation is in the upfront availability of data on interactions with users. This is not surprising, as it is natural to use existing interactions for evaluation when they are available already. The second biggest difference between the groups is whether safety is mentioned as a concern. Relatively, studies that refrain from a realistic evaluation mention safety concerns almost twice as often as studies that do a realistic evaluation. The third biggest difference can be found

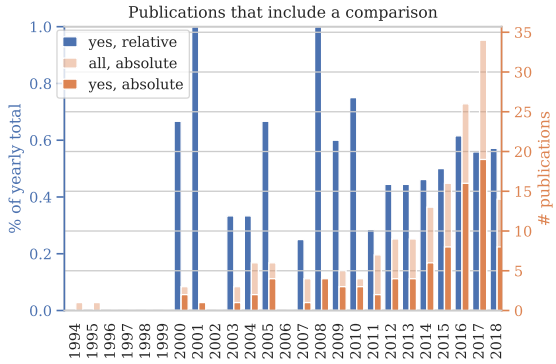


Figure 2.11: Number of papers that include any comparison between solutions over time.

in availability of user models. If a model is available, user responses can be simulated more easily. Privacy concerns are not mentioned frequently, so little can be said on its contribution to a lacking realistic evaluation. Finally and surprisingly, the ease of sampling interactions is comparable between studies with a realistic and without realistic evaluation.

Figure 2.11 describes how many studies include any of the comparisons in scope in this survey, that is: comparisons between solutions with and without personalization, comparisons between RL approaches and other approaches to personalization and comparisons between different RL algorithms. In the first years, no papers includes such a comparison. The period 2000-2010 contains relatively little studies in general and the absolute and relative numbers of studies with a comparison vary. From 2011 to 2018, the absolute number maintains it upward trend. The relative number follows this trend but flattens after 2016.

Table 2.6: Comparison of settings with realistic and other evaluation.

	Real-world evaluation		Other evaluation	
	Count	% of column total	Count	% of column total
Total	104	100.0%	63	100.0%
Data on user responses to system behavior are available	57	54.8%	9	14.5%
Safety is mentioned as a concern in the article	14	13.5%	16	25.8%
Models of user responses to system behavior are available	21	20.2%	20	32.3%
Privacy is mentioned as a concern in the article	7	6.7%	2	3.2%
New interactions with users can be sampled with ease	60	57.7%	37	59.7%

2.7 Discussion

The goal of this study was to give an overview and categorization of RL applications for personalization in different application domains which we addressed using a SLR on settings, solution architectures and evaluation strategies. The main result is the marked increase in studies that use RL for personalization problems over time. Additionally, techniques are increasingly evaluated on real-life data. RL has proven a suitable paradigm for adaptation of systems to individual preferences using data.

Results further indicate that this development is driven by various techniques, which we list in no particular order. Firstly, techniques have been developed to estimate the performance of deploying a particular RL model prior to deployment. This helps in communicating risks and benefits of RL solutions with stakeholders and moves RL further into the realm of feasible technologies for high-impact application domains [352]. For single-step decision making problems, contextual bandit algorithms with theoretical bounds on decision-theoretic regret have become available. For multi-step decision making problems, methods that can estimate the performance of some policy based on data generated by another policy have been developed [63, 165, 355]. Secondly, advances in the field of deep learning have wholly or partly removed the need for feature engineering [90]. This may be especially challenging for sequential decision-making problems as different features may be of importance in different states encountered over time. Finally, research on safe exploration in RL has developed means to avoid harmful actions during exploratory phases of learning [114]. How any these techniques are best applied depends on setting. The collected data can be used to find suitable related work for any particular setting [80].

Since the field of RL for personalization is growing in size, we investigated whether methodological maturity is keeping pace. Results show that the growth in the *number* of studies with a real-life evaluation is not mirrored by growth of the *ratio* of studies with such an evaluation. Similarly, results show no increase in the relative number of studies with a comparison of approaches over time. These may be signs that the maturity of the field fails to keep pace with its growth. This is worrisome, since the advantages of RL over other approaches or between RL algorithms cannot be understood properly without such comparisons. Such comparisons benefit from standardized tasks. Developing standardized personalization datasets and simulation environments is an excellent opportunity for future research [159, 206].

We found that algorithms presented in literature are reused infrequently. Although this phenomenon may be driven by various different underlying dynamics that cannot be untangled using our data, we propose some possible explanations here without particular order. Firstly, it might be the case that separate applications require tailored algorithms to the extend that these can only be used once. This raises the question on the scientific contribution of such a tailored algorithm and does not fit with the reuse of some well-established algorithms. Another explanation is that top-ranked venues prefer contribu-

tions that are theoretical or technical in nature, resulting in minor variations to well-known algorithms being presented as novel. Whether this is the case is out of scope for this research and forms an excellent avenue for future work. A final explanation for us to propose, is the myriad axes along which any RL algorithm can be identified, such as whether and where estimation is involved, which estimation technique is used and how domain knowledge is encoded in the algorithm. This may yield a large number of unique algorithms, constructed out of a relatively small set of core ideas in RL. An overview of these core ideas would be useful in understanding how individual algorithms relate to each other.

On top of algorithm reuse, we analyzed which RL algorithms were used most frequently. Generic and well-established (families of) algorithms such as Q-learning are the most popular. A notable entry in the top six most-used techniques is inverse reinforcement learning (IRL). Its frequent usage is surprising, as the only viable application area of IRL under a decade ago was robotics [180]. Personalization may be one of the other useful application areas of this branch of RL and many existing personalization challenges may still benefit from an IRL approach. Finally, we investigated how many RL models were included in the proposed solutions and found that the majority of studies resorts to using either one RL model in total or one RL model per user. Inspired by common practice of clustering in the related fields such as e.g. recommender systems, we believe that there exists opportunities in pooling data of similar users and training RL models on the pooled data.

Besides these findings, we contribute a categorization of personalization settings in RL. This framework can be used to find related work based on the setting of a problem at hand. In designing such a framework, one has to balance specificity and usefulness of aspects in the framework. We take the aspect of 'safety' as an example: any application of RL will imply safety concerns at some level, but they are more prominent in some application areas. The framework intentionally includes a single ambiguous aspect to describe a broad range 'safety sensitivity levels' in order for it to suit its purpose of navigating literature. A possibility for future work is to extend the framework with other, more formal, aspects of problem setting such as those identified in [304].

3

Reinforcement Learning for Personalized Dialogue Management

Language systems have been of great interest to the research community and have recently reached the mass market through various assistant platforms on the web. Reinforcement Learning methods that optimize dialogue policies have seen successes in past years and have recently been extended into methods that *personalize* the dialogue, e.g. take the personal context of users into account. These works, however, are limited to personalization to a single user with whom they require multiple interactions and do not generalize the usage of context across users. This work introduces a problem where a generalized usage of context is relevant and proposes two Reinforcement Learning (RL)-based approaches to this problem. The first approach uses a single learner and extends the traditional POMDP formulation of dialogue state with features that describe the user context. The second approach segments users by context and then employs a learner per context. We compare these approaches in a benchmark of existing non-RL and RL-based methods in three established and one novel application domain of financial product recommendation. We compare the influence of context and training experiences on performance and find that learning approaches generally outperform a handcrafted gold standard.

Based on [P5]:
Floris den Hengst, Mark Hoogendoorn, Frank van Harmelen and Joost Bosman
Reinforcement Learning for Personalized Dialogue Management
International Conference on Web Intelligence 2019

3.1 Introduction

The use of language by machines has been one of the central challenges in Artificial Intelligence since its initiation as a field of research [230, 366]. Decades of research have advanced the state-of-the-art to such an extent that major consumer-facing web platforms currently offer text- and voice-based 'assistant' capabilities, such as Tencent's WeChat, Microsoft's Cortana, Google's Assistant etc. These platforms have made access to the web through dialogue ordinary. Although such platforms offer high-quality Automatic Speech Recognition (ASR), Natural Language Understanding (NLU) and audio synthesis modules, Dialogue Management (DM) modules are typically handcrafted and require many non-trivial decisions in design and implementation. *Learned* DM based on the formalism of Partially Observable Markov Decision Processes (POMDPs) has shown promising results in task-oriented dialogue systems, both in simulation and real-life settings [116, 301, 400].

Personal context is understood to be fundamental to efficient human-human communication [30]. As a consequence, recent works have addressed the usage of personal context in DM. For example, [217, 241, 356] used previous interactions with a user to directly estimate that users' preferences and then used these estimates in policy optimization. An alternative approach based on transfer learning was presented in [49]. It requires a similarity metric and weighting regime and performance degrades when these are not available. None of these methods generalize the usage of context across users and none of them leverages information available prior to some users' first interaction with the system.

We propose two approaches that optimize the DM policies using personal context. Both approaches are based on the POMDP formalism of learned DM. The first approach consists of extending the POMDP state space with features that describe the personal context of the user. The DM module automatically learns how to use this information for both groups, i.e. it learns the task at hand and segmentation of users simultaneously. This approach allows for personalization to emerge gracefully, e.g. only when enough data is present and when the user model is sufficiently informative for personalization. We compare this approach with a method that explicitly segments users and then uses a learner per user segment. The segmentation of interactions with different user groups mitigates the issue of a 'mixed' signal but leaves less experiences to learn from per learner.

To test our approaches, we extend an existing benchmark for POMDP-based statistical DM for recommendation in three ways [50]. Firstly, we add a novel recommendation task in the financial domain. Here, different user groups have different familiarity with products and specify their preferences at different levels of detail as a result. Secondly, we change the user simulator in the benchmark to reflect this scenario. Thirdly, we add three non-POMDP based approaches to the benchmark: a randomized approach, an approach with a task-specific heuristic and a state-of-art approach based on entropy minimization [391]. To the best of our knowledge, this comparison between POMDP and

non-POMDP based approaches on task-oriented dialog management is novel.

We use the extended benchmark to investigate when each approach is suitable for personalized DM and we investigate the impact of available data to the achieved level of personalization. We first introduce and formalize the recommendation task in Section 3.2 and survey related work in Section 3.3. Next, we introduce the generic approach to RL for DM and then introduce our extensions in Section 3.4. The experimental setup consists of recommendation in existing and novel domains, a user simulator for personalized DM and a benchmark of POMDP and non-POMDP algorithms, is introduced in Section 3.5. After describing and analyzing the results in Section 3.6, we conclude with a discussion in Section 3.7.

3.2 Task Description

This work addresses DM in task-oriented dialogue systems. These systems aim to solve a task by interacting with the user in a conversational style. A popular task for these systems is to recommend a suitable item for a user. The system elicits user preferences or constraints during a dialogue and recommends items from a given item database. We introduce this task formally.

The task addressed in this chapter can be formalized as a q-ary two-player interactive search game [267]. In these game, the goal of one player, dubbed Questioner, is to find a target subset $X_{target} \subseteq X = \{x_1, \ldots, x_n\}$ out of a universe of items X of size n by asking questions to the other player, the Responder. In this case, each $x_i \in X$ consists of a vector of values $\langle x_{i1}, \ldots, x_{im} \rangle$ for features $\{f_1, \ldots, f_m\}$. X_{target} is identified by a set of constraints C, in the form of the desired value c_j for some feature f_j. We assume $\forall\ c_j \in C, \forall\ x_i \in X_{target}, \forall\ x_{ij} \in x_i : x_{ij} = c_j$. Each c_j eliminates a part of the search space. We use C_t to denote the set of constraints at game turn t and X_{C_t} to denote the corresponding candidate item set.

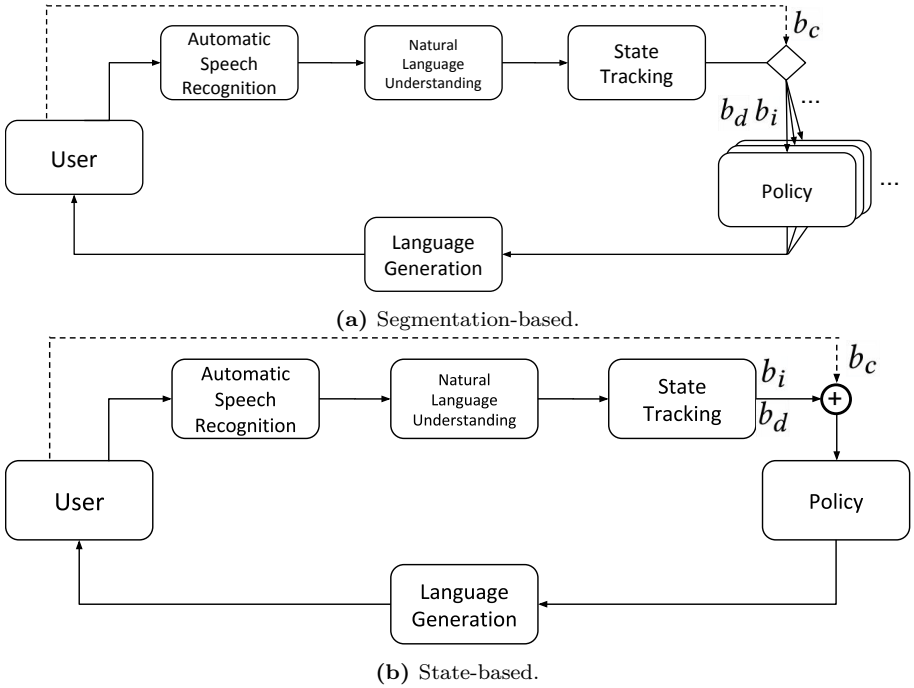
Both the typical q-ary search game and our variation are generalizations of the Rényi-Ulam game (RU game), also known as the binary search game or the parlour game '20 questions'. In RU games, Questions are limited to confirmation of a single constraint, i.e. they are all of the form '$c_j \in C$?' In this format, the optimal question halves the candidate item set X_{C_t} in the optimal case. In our setting, however, the optimal decrease in candidate item set size depends on the distribution of values for all f_j's in X_{C_t}. The Questioner may use knowledge about these distributions in selecting a f_j to ask a constraint for. We therefore include a policy that uses knowledge about the distribution of values in all f_j's as a search heuristic. More so, the Responders' tendency to provide constraints for a feature f_j may not be distributed uniformly in realistic settings. A Questioner with access to past plays may use this experience to estimate the likelihood of a constraint for a feature being present to find an item more efficiently. We therefore include approaches that can leverage experience into our benchmark, see Section 3.5.3 for details.

3.3 Related Work

Most approaches to personalizing dialogue systems can be categorized as learning-based or rule-based. We provide a brief overview of approaches in both categories. An example of a rule-based approach can be found in [126] and [356]. This system uses a model of user preferences for constraints c_j to weigh factors that determine similarity of a user query to the items in X. The DM policy is handcrafted, which typically entails many nontrivial decisions that can seriously impact system performance [205]. More recent examples, such as [20, 178, 320] collect user-related facts in a knowledge graph. These facts are then used to personalize hand-crafted response templates. These approaches focus on personalized natural language generation and have handcrafted DM modules.

Learning-based approaches, on the other hand, optimize the DM policy using experiences with real or simulated users. A conversational shopping recommender is described in [217]. It requires multiple interactions with a specific user and has a query-response interaction style. An example with a natural language interaction style based on transfer learning can be found in [49]. It initializes a policy for the target user by training on data from interactions with similar users. The authors find that it is beneficial to include data from dissimilar users, albeit with lower weights, as this results in better coverage of the state space during training. A drawback of the approach is that it requires a suitable similarity metric. A transfer learning-based approach that does not suffer from this drawback is introduced in [241]. A policy is optimized using a global optimization criterion and all available experiences. Next, the optimization criterion is extended with user-specific slot-value preference estimates which are updated in subsequent interactions. This approach only adapts to individual users in terms of slot-value preferences and requires multiple interactions with a single user. A third transfer learning-based approach is presented in [120]. The selection of experiences to train the model on for a specific user is cast as a multi-armed bandit problem. Finding a source of experiences out of all n users, however, requires at least n bandit trials. This limits applicability to scenarios with a small number of users.

None of the approaches discussed so far leverage information external to the conversation, e.g. context, to optimize the dialogue policy. In non-conversational recommendation, however, numerous works rely on the users' personal contexts. As a full survey is out of scope for this chapter, so we focus on generic trends instead. Recommender systems are typically classified as content-based, collaborative filtering or a hybrid of these two. Content-based recommender systems 'exploit the user profile to suggest relevant items by matching the profile representation against that of items to be recommended' and thus rely on the users' personal context [263]. Collaborative filtering selects items for recommendation by looking at past consumption patterns by similar users and personal context can be used to determine similarity of users [3, 172, 213]. Out of these approaches, contextual bandit methods are specifically related to this work. These methods aim to determine how elements



(a) Segmentation-based.

(b) State-based.

Figure 3.1: RL-based approaches to personalized DM.

of personal context affect relevance of items through subsequent interactions with users [197]. These methods, however, are not suitable for conversational settings as they do not take sparsity of rewards and the sequential nature of these settings into account.

3.4 Approach

This section describes two novel approaches to personalized DM for the interactive recommendation task described in Section 3.2. First, the formalism of Partially Observable Markov Decision Problems is described and it is explained how it it can be applied to DM for the interactive recommendation task.

3.4.1 RL for DM

State of the art statistical dialogue systems cast DM as a Partially Observable Markov Decision Problem (POMDP) [301] [388]. A POMDP is a generalization of a Markov Decision Process where the true state is not directly observable, but must be estimated through observations. In dialogue systems, the source of uncertainty about the true state stems from errors in Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU) modules. The

POMDP is defined as $M = \langle S, A, T, R, \Omega, O \rangle$ where $S \in \{s_1, \ldots, s_n\}$ denotes a finite set of partially observable states representing user intentions and dialogue history, $A \in \{a_1, \ldots, a_m\}$ is a finite set of actions representing system responses, $T\colon S \times A \times S \rightarrow [0, 1]$ is a probabilistic transition function over states and $R\colon S \times A \rightarrow \mathbb{R}$ denotes a reward function based on number of turns and accuracy of recommendation, $\Omega \in \{o_1, \ldots, o_l\}$ is a finite set of observations available to the system, and $O\colon \Omega \times A \times S \rightarrow [0, 1]$ denotes a probabilistic function over observations, actions and states. The true state s is unavailable to the agent, only observations Ω are.

The dependence of O on Ω and A makes the decision process non-Markovian and thus unsuitable for standard RL algorithms. The Markovian property can be regained, however, by maintaining a Bayesian belief over S and substituting the original state space with this belief space. This substitution leaves us with a continuous MDP with an input space $B \in \{b_1, \ldots, b_o\}$ with dimensionality $|S| - 1$, which is too complex for most practical purposes. In practice, however, the belief space can be significantly reduced in size by splitting it into factors and assuming mutual independence between factors. In dialogue systems aimed at the interactive recommendation task from Section 3.2, the belief space can be split into a factored belief space B' consisting of dialogue history belief b_d and a user intention belief b_i. The dialogue history b_d describes, for example, whether the system has already recommended an item x_i or requested a constraint for feature f_j. The user intention belief describes preferences of the user w.r.t. the product database. Maintaining this state is a challenge in itself, but outside of the scope of this work. See [387] and [147] for overviews. As B is replaced by B' and not used anymore, we denote B' as B from here on.

Constructing the POMDP involves some design decisions based on the task at hand. Specifically, A should contain actions that are useful or necessary for the agent to achieve its task. For the interactive recommendation task the agent plays the part of Questioner. The available utterances should thus at least reflect requesting a constraint for each feature f_j and recommending an item. Additional actions can make the dialogue more natural and efficient, such as confirmation questions of the form '$c_j \in C$?' and selection questions of the form '$c_j \in C$ *or* $c_{j'} \in C$?'.

Besides a suitably defined A, the POMDP should be constructed with an R that reflects the goal of the task at hand. This work is based on a benchmark further described in Section 3.5. In the benchmark R is defined as

$$20 * acc(X_{target}, \langle a^1, \ldots, a^l \rangle) - l \tag{3.1}$$

for a given X_{target} and trajectory of system actions $\langle a^1, \ldots, a^l \rangle$ of length l. $acc()$ returns 1 if the trajectory contains a recommendation action for an item $x_i \in X_{target}$ and 0 otherwise. The goal is to find the optimal function $\pi^* : B \rightarrow A$ that maximizes the expected sum of discounted future rewards

$$\pi^*(b) = \arg\max_a Q^{\pi^*}(b, a), \forall b \in B, \forall a \in A \tag{3.2}$$

where

$$Q^{\pi^*}(b,a) = E_{\pi^*}\Big\{ \sum_{k=0}^{\infty} \gamma^k r^{t+k+1} | b^t = b, a^t = a \Big\} \tag{3.3}$$

and $\gamma \in [0,1]$ is a factor weighing future rewards and b^t and a^t are future beliefs and actions.

3.4.2 Personalized Dialogue Management

We present two approaches to DM using personal context of the user based on the formalism described. Figure 3.1 provides an overview of the two methods. Both use a vector describing the agents' belief of *personal context* b_c of the user to optimize the dialogue for specific users. This may include any available information about the user that may aid in policy optimization. Examples of context include demographics, purchase history and previous interactions. Note that context need not be constant during or in between dialogues. This section describes how context is used in both methods.

The method in Figure 3.1a is based on segmentation of the user population by context. It assumes a function $M : B_c \to G$ that maps agent beliefs on user contexts $B_c \in \{b_{c_1}, b_{c_2}, \ldots, b_{c_n}\}$ to segments $g \in G$ (g for 'group'). A separate policy $\pi_g(b_d, b_i)$ is maintained that exclusively interacts with contexts b_c for which $M(b_c) = g$. As the policy interacts with user contexts in a single segment, it learns a policy optimal for that segment using only beliefs on dialogue history b_d and user intentions b_i. The context b_c is not available to the policy. A benefit of this approach is the absence of negative transfer between segments: behaviors suitable to only a particular segment of users are only learned by that segments' policy and will not be considered suitable by policies serving the other segments. On the other hand, there cannot be any positive transfer either: each policy is exposed to less interactions which may result in poor belief state space coverage and degraded performance. Furthermore, it may be nontrivial to find a suitable segmentation function M as this involves finding an unambiguous context representation and determining the number of segments.

The method in Figure 3.1b does not suffer from these drawbacks. It consists of concatenating beliefs on dialogue history b_d, user intentions b_i and context b_c. The resulting belief vector is then used as input to a single policy $\pi_p(b_d, b_i, b_c)$ for the entire user population. An algorithm that optimizes π_p now jointly learns DM and the usage of context therein. This allows for the learner to only use context when it is beneficial and liberates us from defining segmentation or similarity criteria. The composed learning task, however, may be significantly more challenging as users from different segments may have conflicting desires. This might lead to a form of negative transfer that the algorithm optimizing π_p has to be robust to which may require more training data.

Domain	# Items	Group 1 & 2	Group 2 only
CR	110	price range	area, food
SFR	271	price range, allowed for kids, good for meal	area, near, food
LAP	123	utility, price range, weight range, warranty, is for business computing	family, processor class, sys memory, platform, drive range, battery rating
FIN	14	minimum age, purpose, account	name, insurance, max. duration, min. duration, max. principal, min. principal

Table 3.1: Usage of slots for constraints for the two user groups. Group 1 denotes users unfamiliar with the domain or 'laypersons' while Group 2 denotes users experienced in the domain or 'experts'. Expert users always use three constraints, whereas layperson users have between one and three constraints.

3.5 Experimental Setup

The goal of this chapter is to evaluate the proposed approaches for personalized dialogue management. We split this goal into the following research questions. In a personalized DM task,

- Q1 when do learning-based algorithms outperform handcrafted algorithms?
- Q2 when do belief state-based approaches outperform segmentation-based approaches?
- Q3 how well do existing approaches generalize to the novel domain of financial product recommendation?

Regarding these research questions, we hypothesize:

- H1 learning-based approach only outperform handcrafted approaches in the presence of preprocessing errors.
- H2 belief state-based approaches perform comparable to or better than segmentation-based approaches.
- H3a in the new domain, learning-based approaches perform comparable to existing domains.
- H3b in the new domain, handcrafted approaches perform worse than in existing domains.

The experimental setup is based on a benchmark suite for task-oriented dialog management [50]. The suite includes a user simulator, a dialog management

module and DM algorithms. The benchmark further consists of recommendation tasks in three domains: recommendation of restaurants in Cambridge (CR), of restaurants in San Francisco (SFR) and laptops (LAP), we refer to [50] for details. We extend this benchmark in three ways. Firstly, we add a new domain of recommending financial products. Secondly, we extend the user simulator to include context. Finally, we add our proposed algorithms and additional non-POMDP-based algorithms to the benchmark.

3.5.1 Recommending Financial Products

The financial domain is an interesting addition as it is different from domains currently in the benchmark: the number of interactions with a single user is typically limited, there may be large gaps in between interactions and user intentions are typically not constant over interactions. It is, for example, unlikely that a single customer needs multiple recommendations based on an intention to finance a car purchase. This renders approaches that require multiple interactions with a single user or that rely on direct estimation of user preferences inapplicable.

A second particularity of this domain is that different users have different familiarity with products. As a result, users in this domain have differing preferences and ability to express them. For example, customers that have a car loan will be more familiar with technicalities of secured loans and therefore be more capable of expressing their preferences for similar loans in detail. Such differences are common in domains with complex products, such as the financial, technology and automotive domains. Although the exact formulation of context is not the focus of this work and may vary per domain, we consulted with domain experts in the financial domain on contextual factors currently used in determining how to communicate with users across various channels. These domain experts indicate that one of the major factors in communicating about a product is whether the user consumes a product from the same product category.

		Entropy		**POMDP**			
	RQ	EMDB	EMDM	HDC	RL_v	RL_s	RL_{bs}
Task-specific		v					
NLU/DST-error aware				v	v	v	v
Adaptive			v		v	v	v
Uses context						fixed	adaptive

Table 3.2: Overview of qualities of approaches. RL_v, RL_s and RL_{bs} describe the vanilla, segmentation-based and belief-state based versions of *GP*, *A2C*, *DQN* and *eNAC*.

Differences with other domains are not limited to typical interaction patterns, however: the item set X is distinctive in this novel setting as well. This

item set was developed using using well-known ontology engineering practices and evaluated with domain experts [10, 255]. The resulting item set consists of 14 products and 13 features. Nine out of these can be used as a constraint by the user, see Table 3.1 for an overview. All other slots are only used to inform the user about the product and not relevant to the recommendation task. The number of values for all constraint features is 64. When compared to the existing domains in literature, the novel FIN domain has a relatively small item set and relatively large number of constraint-slots. We add this item set as an 'ontology' to the Pydial benchmark for DM systems [50] which is described in the next Sections in more detail.

3.5.2 User Simulator

We adapt the user simulator in the benchmark as described in [310] to reflect the scenario from the previous section. A full description of this simulator is out of scope and we limit ourselves to the main concepts before moving on to the extensions. In the simulator, actions by the simulated user are conditioned on the dialogue so far and on behavior parameters and includes an error model for ASR and NLU modules. Parameters for all of these have been tuned using data from experiments with real users, for details see [310]. Behavior parameters are sampled at the start of each dialogue and according to distributions that have been set in user profiles so that each dialogue is with a user with individual behavior characteristics. Similarly, up to three constraints c_j are sampled randomly for each new simulated user. Additionally, heuristics to constrain the action space can be enabled or disabled. These *action masks* make part of the action space unavailable and ease the learning task. A combination of user model, error model and availability of action mask is denoted as an 'environment'. In total, the benchmark we use includes six different environments [50].

We extend the tuned simulator with user context to reflect the scenario from the previous section. Two user groups are modelled. The first group represents 'laypersons' that express constraints for specific slots only; the second group represents knowledgeable users that express constraints for all slots. All slots and their usage per group are listed in Table 3.1. The usage of slots between groups for the FIN domain has been set after consultation with domain experts. For the CR, SFR and LAP domains, these are set to allow for a comparison of approaches across settings.

We add a b_c to describe the user context and add per-slot constraint usage parameters to the simulator. Specifically, b_c is a vector of two values, describing the belief on the user having experience in the domain or not. Although our approach facilitates a wide range of values, we here limit ourselves to the case of fully certain upfront knowledge, i.e. $b_c \in \{0, 1\}^2$. We assume that interactions with both types of users are equally likely.

3.5.3 Algorithms

We evaluate our approach using all algorithms presented in the benchmark from [50] and measure per-dialogue rewards according to equation 3.1 in Section 3.4.1 across 10 random seeds with 4000 training and 500 test dialogues each. The benchmark contains one handcrafted policy, *HDC*, and four RL-based algorithms: *GP* for GP-SARSA, *A2C*, *eNAC* and *DQN*. All of these algorithms are based on the POMDP formalism introduced in Section 3.4.1. *GP* is a data-efficient nonparametric value-based approach that uses Gaussian Processes to estimate $Q^\pi(b, a)$ from equation 3.3 [116]. *DQN* similarly estimates these Q values using a neural network, i.e. it is a parametric approach [332] [371]. *A2C* and *eNAC* are parametric algorithms that estimate the policy $\pi(b)$ as defined in equation 3.3 directly, where *A2C* estimates $Q(b, a)$ additionally [94]. We refer to [50] for more detail on these algorithms. We include vanilla versions of the learning algorithms, versions based on segmentation and versions based on an altered belief-state and denote these by $\cdot_v$, $\cdot_s$ and $\cdot_{bs}$ subscripts respectively.

We further extend the benchmark with three non-RL-based algorithms.[1] The algorithms were selected based on the task formalization of Section 3.2 and to enable a comparison of learning algorithms versus handcrafted algorithms. Specifically, we add a randomized baseline, an algorithm with a search heuristic and a state-of-art learning method from [391]. This last method keeps a history of successful dialogues as trajectories of user utterances u and system actions $\langle u^1, a^1, ..., u^\ell, a^\ell \rangle$ up to a successful recommendation a^ℓ. During a dialogue $\langle u^1, a^1, \ldots, u^t \rangle$, the system selects the action a^t that minimizes the entropy of all past successful recommendations a^ℓ, breaking ties with a random selection. We denote this approach with *EMDM* for 'Entropy Minimization Dialog Management'.

The two remaining non-POMDP-based algorithms are a randomized baseline and a baseline that uses information about the product database. The randomized baseline randomly asks for constraints on feature f_j until there are no differentiating features in X_{C_t} and then recommends some item $x_i \in X_{C_t}$ randomly. We denote this baseline with *RQ* for 'Random Question'. The second baseline has the same strategy for recommending an item, but differs in selecting f_j. Given the current X_{C_t}, it selects the f_j with the highest entropy in the candidate item set X_{C_t} and requests the user preference for it. This is a task-specific approach that uses a entropy as a heuristic to search the item set X_{C_t} efficiently. We denote this benchmark as *EMDB* for 'Entropy Minimization DataBase'. All non-POMDP-based approaches, i.e. *RQ*, *EMDM* and *EMDB*, have no way of dealing with errors from the ASR and NLU modules in Figure 3.1. The output of these modules with the highest confidence score is simply assumed as correct and used as input to these algorithms.

[1] Code: `https://bitbucket.org/florisdenhengst/pydial/commits/tag/web-intelligence-19`

	# Nodes		
Model	Hidden Layer 1	Hidden layer 2	ϵ_s
DQN	300	100	.5
A2C	200	75	.5
eNAC	130	50	.3

Table 3.3: Hyperparameters for neural network based approaches.

3.5.4 Environment and Hyperparameters

All experiments were run on Intel Xeon Silver 4110 Processors using Python version 2.7.9, TensorFlow version 1.12.0, NumPy version 1.15.4 and SciPy version 1.2.0. Ten different random seeds ranging from zero to ten were used. Hyperparameters were set as in [50], we repeat them here. For the *GP* algorithm, a linear kernel was used on the state space and a Kronecker delta kernel was used on the action space. The 'scale' variable of these determines the rate of exploration and was set to 3.

DQN, *A2C* and *eNAC* use an ϵ-greedy exploration strategy during training where ϵ is linearly scaled between ϵ_s and 0.05 in training, i.e. for the 4,000 dialogues. Exploration was turned off during evaluation. See Table 3.3 for values of ϵ_s and network architecture for the neural network based approaches. For these, the architecture consisted of three layers of fully connected feedforward of varying sizes. The Adam optimizer was used for training with an initial learning rate of 0.001. We refer to the code repository for further details on the hyperparameters.

3.6 Results

In this section, we describe the results with respect to the research questions from Section 3.5. Table 3.4 lists all results.

Q1 Figure 3.2a shows the performance of the best algorithms in an environment where ASR/NLU errors are absent. According to hypothesis H1, we expected the *HDC* and *EMDB* algorithms to outperform learning algorithms. We analyse the performance of these algorithms per domain. The CR domain contains relatively little slots and groups are similar. The task-specific *EMDB* algorithm moderately outperforms learning-based approaches GP_s and DQN_s which in turn outperform the *HDC* algorithm. Moving to the FIN domain, DQN_s and GP_s outperform *HDC* due to the large difference between groups. We analyze the poor results of *EMDB* in this novel domain below (Q3). In the LAP domain, the *EMDB* algorithm performs the worst out of the selected algorithms. This domain has a large number of slots hence there is likely to be a differentiating feature f_j that will be selected according to *EMDB*. The *EMDB* algorithm thus keeps on asking for new f_j, even when the user has already listed all of their requirements. Comparing *HDC* with learning-based approaches in this domain, it performs comparable to DQN_s and GP_s. The

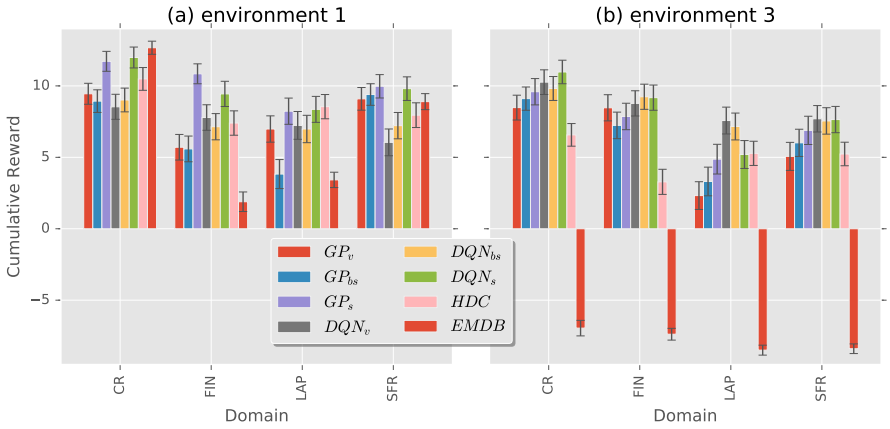


Figure 3.2: Average reward per dialogue in test set for environments without (a) and with (b) ASR/NLU errors.

reason for this may be that this is a relatively challenging learning task which limits the benefits of personalization. The SFR domain has a relatively large item set X and a moderate number of slots. The search heuristic of *EMDB* works as expected here and GP_s and DQN_s moderately outperform handcrafted approaches. Overall, we find that –in contrast to H1– learning-based approaches perform comparable or better than both handcrafted approaches, *even in the absence of ASR/NLU errors.*

We now compare these families of approaches in an environment with ASR/NLU errors in Figure 3.2b. In this setting, the gold standard *HDC* algorithm degrades more than learning approaches, further supporting the benefits of learning approaches in a scenario with different user groups. The difference can be explained by *HDC*'s response to an unclear answer for some slot: it requests the user to confirm the most likely value as recorded by the ASR/NLU modules. Such a request will not further the dialogue if that particular slot does not contain a constraint for the user. The *HDC* algorithm does not take this into account, whereas learning approaches can adapt to the laypersons' inability to informatively respond after such a confirmation request and ask for other constraints first. The *EMDB* algorithm cannot handle uncertainty from ASR/NLU outputs. It assumes the most likely preference as indicated by ASR/NLU modules. This assumption is occassionally incorrect and generally ruins *EMDB*'s performance.

Q2 In contrast to hypothesis H2, performance of belief state- and segmentation-based personalization approaches vary across domains, environments and used learning algorithms. For the *GP* algorithm, segmentation generally outperforms vanilla and belief-state based approaches in both environments. This suggests that *GP* suffers less from lack of training data as a result of segmentation, which is in line with earlier findings that *GP* is a data efficient algorithm [116]. The performance of this algorithm relies on the

chosen kernel. In the benchmark, a linear kernel is used. This kernel assumes a linear relation between $Q^\pi(b, a)$ and the belief state b. We briefly analyze this linearity assumption by considering two similar belief states b that only differ in the belief on user group membership for the current user b^c. The linearity assumption implies that some favorable action for the first group is unfavorable for the other group. This assumption clearly does not hold for some actions, e.g. requesting some f_j that is used by both groups.

For *DQN*, some negative effects of segmentation can be seen in cases with a complex learning problem, i.e. in environments with ASR/NLU errors and in domains with a large state space. These negative effects can be mainly seen in domains with larger state spaces LAP and SFR. Regarding the belief state-based approach, results indicate that it performs comparable or slightly better than the vanilla approaches in most configurations. We hypothesized that this approach would learn to exploit differences in user population without suffering from the drawback of limited training data as in the segmentation-based approach. Although our findings indicate that the latter is generally the case, the benefits of personalization diminish for more complex learning problems in environments 4-6. A possible explanation for this is that the algorithms' hyper-parameters, specifically the neural network architecture for *DQN* and kernel for *GP*, were not optimized to the personalization setting.

Q3 Figure 3.3 shows how POMDP-based approaches hold over various domains in all included environments. We omit non-POMDP-based approaches here due to their poor performance in environments 3-6. When comparing the novel FIN domain, the gold standard *HDC* is outperformed by all considered learning algorithms. The learning algorithms generalize to the new domain. The *HDC* policy was handcrafted for the other four domains and does not transfer well to a novel domain with different characteristics. To analyze the results of *EMDB* in the FIN domain, we consider again Figure 3.2. In the FIN domain, the item set X is small which makes the search heuristic on which *EMDB* relies inapplicable. These results are in line with hypotheses H3a and H3b.

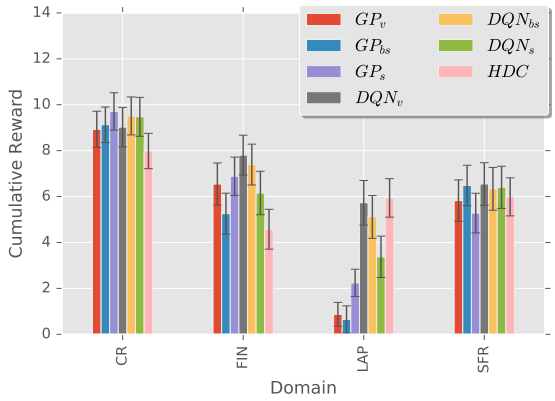


Figure 3.3: Per-dialogue reward of selected algorithms in test set, averaged over all environments.

env	Error %	Action Masks	User Model	domain	$A2C_v$	$A2C_{bs}$	$A2C_s$	DQN_v	DQN_{bs}	DQN_s	$eNAC_v$	$eNAC_{bs}$	$eNAC_s$	GP_v	GP_{bs}	GP_s	HDC	RQ	$EMDB$	$EMDM$
1	0	y	normal	CR	12.2	11.6	10.6	8.5	9.0	12.0	1.4	6.5	10.6	9.4	8.9	11.7	10.5	**12.7**	**12.7**	-4.7
				FIN	**10.9**	8.5	7.2	7.8	7.1	9.4	4.1	0.8	8.0	5.7	5.6	10.8	7.4	2.3	1.9	-12.3
				LAP	5.9	4.1	0.6	7.2	7.0	8.4	7.5	7.9	5.9	7.0	3.8	8.2	**8.5**	4.2	3.4	-14.0
				SFR	6.4	6.4	5.1	6.0	7.2	9.8	5.2	5.0	8.3	9.1	9.4	**10.0**	8.0	9.2	8.9	-8.8
2	0	n	normal	CR	2.8	2.3	2.2	11.8	11.2	11.3	-4.4	-3.8	3.2	11.7	11.6	11.7	11.9	**12.7**	**12.7**	-4.7
				FIN	2.8	3.2	3.4	10.7	9.8	5.7	-3.2	-2.2	3.8	8.1	5.4	6.7	8.5	2.3	-10.0	-12.3
				LAP	-2.7	-2.4	-2.5	6.3	5.7	1.8	-3.3	-3.7	-0.1	-1.0	-0.9	-0.9	10.3	4.2	3.4	-14.0
				SFR	-0.8	0.1	-1.6	9.4	7.4	7.4	5.0	5.1	0.2	8.8	8.6	5.4	10.3	9.2	8.9	-8.8
3	15	y	normal	CR	8.2	8.1	7.8	10.3	9.8	**11.0**	7.0	8.0	10.0	8.5	9.1	9.6	6.6	-7.4	-7.0	-5.3
				FIN	6.2	5.4	3.2	8.8	**9.2**	**9.2**	4.6	7.0	6.8	8.5	7.2	7.9	3.3	-7.8	-7.4	-12.5
				LAP	-1.3	-0.9	-2.2	**7.6**	7.2	5.2	5.7	5.5	4.6	2.3	3.3	4.9	5.3	-8.7	-8.5	-14.3
				SFR	0.8	1.1	0.1	**7.7**	7.5	7.6	6.3	7.1	4.2	5.1	6.0	6.9	5.2	-8.4	-8.4	-9.7
4	15	n	normal	CR	2.4	2.6	1.4	**10.2**	9.5	7.1	0.9	1.7	2.9	9.6	9.7	8.9	6.6	-7.4	-7.0	-5.3
				FIN	3.3	4.2	1.3	**9.6**	7.1	4.6	-1.0	-1.0	4.3	6.4	5.2	5.4	3.3	-7.8	-7.4	-12.5
				LAP	-3.0	-3.1	-2.7	4.6	3.4	-0.1	-3.8	-0.3	-2.5	-1.1	-1.0	-1.0	**5.3**	-8.7	-8.5	-14.3
				SFR	-1.0	0.2	-1.8	5.2	**6.7**	4.3	-1.1	2.0	0.9	4.6	4.7	2.5	5.2	-8.4	-8.4	-9.7
5	15	n	unfrndly	CR	6.6	4.6	4.8	7.0	**9.7**	8.3	4.9	7.7	7.6	7.5	8.3	8.9	6.7	-7.5	-7.5	-5.5
				FIN	2.2	2.1	1.6	6.2	**7.2**	4.1	4.4	5.5	5.1	5.3	4.6	5.6	2.5	-7.8	-7.5	-12.8
				LAP	-3.3	-2.0	-3.1	3.7	**4.1**	1.9	1.8	1.8	0.5	-0.0	-0.1	1.7	3.0	-8.6	-8.4	-14.6
				SFR	-2.1	-0.1	-1.1	**5.3**	4.6	4.6	2.3	3.3	4.1	3.8	3.6	3.5	3.7	-8.4	-8.4	-10.3
6	30	y	normal	CR	4.2	4.2	4.8	6.4	**7.8**	7.2	6.2	7.1	7.2	6.8	7.1	7.3	5.6	-4.7	-4.7	-5.8
				FIN	0.6	0.2	0.5	3.7	3.8	3.8	3.8	4.8	**5.6**	5.2	3.5	4.9	2.5	-7.6	-7.0	-12.6
				LAP	-2.8	-2.6	-2.3	**4.9**	3.4	3.1	3.2	3.3	2.0	-2.0	-1.2	0.4	3.2	-9.3	-8.8	-14.5
				SFR	1.6	-1.8	-0.5	**5.7**	4.6	4.6	4.2	4.7	4.9	3.6	2.4	3.5	3.5	-8.3	-8.0	-9.7
mean					2.51	2.34	1.54	7.28	7.09	6.35	2.57	3.49	4.52	5.54	5.38	6.03	6.12	-2.92	-3.37	-10.38

Table 3.4: Average reward per dialogue for test set across environments, domains and algorithms in the benchmark.

3.7 Discussion

In this work, we have proposed two approaches to DM using personal context and evaluated them on various environments, in various domains and using various algorithms. The approaches leverage existing contextual information about a particular user and can offer personalized DM even in the absence of previous interactions with a particular user.

In order to evaluate our approaches, we have extended an existing benchmark for conversational item recommendation with two user contexts and associated behavior patterns. The behavior patterns reflect those found in domains where 'expert' and 'layperson' users have differing knowledge about the available items. Results indicate that learning a dialogue policy is beneficial in settings with differing user behaviors. Notably, the addition of context boosts performance of learned dialogue managers to comparable or higher levels than a handcrafted gold standard and task-specific approaches, even in an environment without noise from preprocessing modules.

We find that performance of learning approaches varies with environment, domain, and algorithm. Specifically, data efficiency could be investigated by increasing the number of training dialogues. Similarly, the applicability of the approaches could be investigated by varying the difference between user groups. Furthermore, varying hyperparameter settings such as neural network architecture and learning rate and more powerful and stable RL algorithms may lead to more the complex behaviors in the new setting such as those in [137]. More experiments are necessary to further investigate performance characteristics for the proposed approaches.

With regards to methodology, we have introduced a case validated by domain experts in the financial domain and added it to an existing benchmark of item recommendation. We have extended a realistic user simulator with additional behavior parameters for all domains in the benchmark to comprehensively test our approaches. Although these additional parameters are suitable to test our approaches technically, they were not sampled from real-world data. Comparing the approaches in real-world settings, such as an evaluation with real users or an evaluation in a configuration where behavior parameters are based on real-world differences between experts and laypersons would be interesting next steps.

Finally, we tested our approaches to the usage of context in a specific case with different user groups with static context information and a constant action space. Our approaches, however, are general and could be applied to various other usages of context to dialogue policy optimization. Especially interesting would be the inclusion of sentiment estimates as in [279]. Together with an extension of the action space, these could aid in making the conversation more natural by conditioning e.g. trust-building system responses on conversation content and context at the same time.

4 Collecting High Quality Dialogue User Satisfaction Ratings with Third-Party Annotators

The design, evaluation and adaptation of conversational information systems are typically guided by ratings from third-party, i.e. non-user, annotators. Interfaces used in gathering such ratings are designed in an ad-hoc fashion as it has not yet been investigated which design yields high-quality ratings. This work describes how to design user interfaces for gathering high-quality ratings with third-party annotators. In a user study, we compare a base interface that consolidates best practices from literature, an interface with clear definitions and an interface in which tasks are separated visually. We find that these interfaces yield annotations of high quality and separation of tasks. We find no significant improvements in quality between User Interfaces (UIs). This work can serve as a starting point for researchers and practitioners interested in collecting high-quality dialogue user satisfaction ratings using third-party annotators.

Based on [P7]:
Mickey van Zeelt, Floris den Hengst and Seyyed Hadi Hashemi
Collecting High Quality Dialogue User Satisfaction Ratings with Third-Party Annotators
Proceedings of the 2020 Conference on Human Information Interaction and Retrieval

4.1 Introduction

Conversational information interfaces have been of interest to research and industry for decades [231]. In more recent years, devices such as smart speakers, phones and in-car systems have made interaction with conversational interfaces common at home, in public spaces and traffic. These interfaces may be developed on cloud-based platforms such as Amazon Alexa and Google DialogFlow. Although these platforms offer high-quality reusable components for subtasks such as converting text to speech and extracting keywords from utterances, the overall quality of the entire interface typically needs to be evaluated per system deployment. Furthermore, quality ratings may be required to adapt and personalize the interface in an online fashion as in Chapter 3.

Although various signals for capturing the quality of conversational interfaces have been studied, user satisfaction is typically the ultimate metric to optimize for. User satisfaction is a subjective measure of the quality of an interaction [176] and a rating of user satisfaction can be acquired from users directly or from third-party annotators. These two types of ratings are considered complementary [367]. Third-party ratings come with the inherent challenge that the annotator does not know the intent of the user. In contrast to user ratings though, they can be acquired at manageable costs in a controlled environment.

Acquiring third-party ratings of high quality remains challenging, though. Previous research frequently gathered third-party ratings with low reliability or agreement between raters [7, 143, 151, 154, 312, 399]. The reasons for this have been studied to some extent. Estimation of and response to the unreliability of raters in e.g. crowdsourcing has in particular received a substantial amount of attention [43, 46, 105, 175, 225, 326]. Other aspects, such as annotation interface design, however, have largely remained unaddressed and interfaces are typically developed in an ad-hoc fashion as a result.

Some works have indicated that definitions aid in collecting high quality annotations [105, 124, 179, 246], whereas others have found that the bias thus introduced yields lower quality annotations [289]. Additionally, it is unclear whether forcing the user to complete a task before starting a subsequent task by vertically positioning UI elements improves quality or harms it. This technique, called *cascading*, was reported to be beneficial by Lin et al. [204] and Gligorov [124] whereas it was thought to complicate the interface by Real et al. [289].

Therefore, the main research question addressed in this work is how to develop a user interface (UI) that facilitates the gathering of high-quality third-party dialogue annotations. In particular, we investigate:

> **RQ1:** How do definitions of "user satisfaction" in the interface affect the quality of annotations?
>
> **RQ2:** Is a cascading user interface preferable over a non-cascading interface with visual separation of tasks?

We design three UIs to investigate these research questions and find that they yield high quality annotations in a user study (n=27).

4.2 Method

In order to answer the research questions from the introduction, three UI variations were designed, developed and evaluated. In the design phase, guidelines were gathered from literature and converted to requirements for all interfaces. In this section, we first outline these requirements. Next, requirements that reflect the research questions from Section 4.1 were added to each variation. We developed a base interface, a variation with clear definitions and a variation with visually separated tasks. We further detail these variations below.

4.2.1 Requirements

From the field of information presentation, we know that the number of tasks in a UI negatively affects quality for each task [105]. Therefore, interfaces should be limited to as little tasks as possible. In the case of conversational interfaces, annotators should read the full conversation and then evaluate user satisfaction. Additional tasks, such as providing confidence scores, should be avoided.

Another means to reduce UI complexity is to limit the number of choices [124, 179, 289]. Considering that user satisfaction is a subjective measure, annotators typically find it challenging to choose between two options. A five-level Likert scale effectively balances the number of choices and expressiveness of the rating [143, 176, 247].

A third means to reduce UI complexity is to simplify any text involved. Text should be presented in short sentences, with key phrases in bold font. Descriptions of input should be structured to contain questions at the end. This increases the likelihood of annotators reading text fully. Text consisting of just a few lines has been found to be less likely to be read fully or even completely when compared to text in a single brief sentence [289].

Annotators develop a better understanding of the task while performing it [261]. These *learning effects* yield unwanted differences between ratings by a single annotator over time. To combat these effects, training is recommended [175, 191]. The training should introduce the interface, concepts and tasks [289, 326].

Duration of the task may also negatively affect quality. Research indicates that fatigue factors have a significant effect on the quality with a peak annotation quality at 30 minutes of participation [46]. Task duration measurements can be performed during pilots of the UI and findings can be used to tune the task size. Additionally, limitations on consecutive participation can be put in place. Finally, total task duration can be logged so that annotations of suspected lower quality may be removed before analysis.

In order to ensure that participants are suitable, participants may have to be filtered up-front. Filters may be based on experience with similar tasks, country of origin and familiarity with the language and cultural context, as well as a qualification test [46, 326]. A test is typically not preferred: it may introduce additional fatigue in annotators and may filter out raters that would

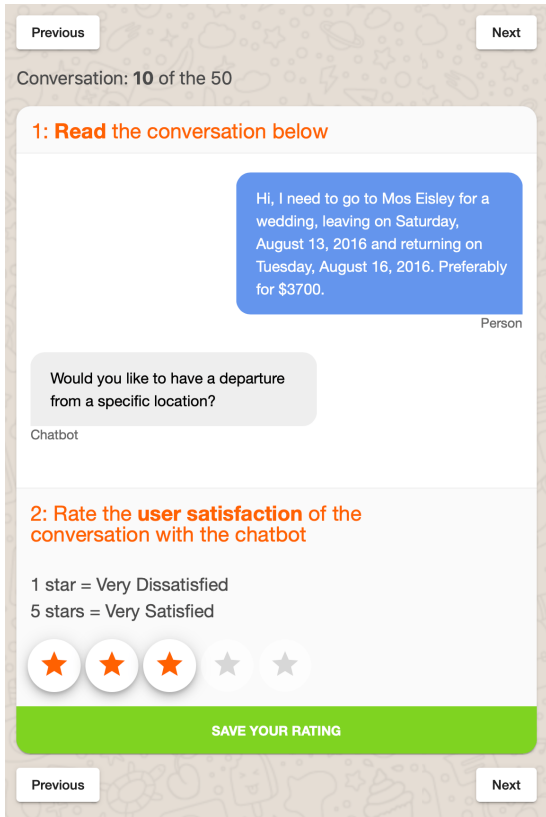


Figure 4.1: UI1: Showing the focus on cascading the elements and keeping the information on the task ambiguous.

provide qualitative ratings after training [46, 105]. Ideally, intrinsic motivation is to be preferred over financial incentives as Finnerty et al. [105] and Mason and Watts [225] indicate that these speed up data collection at the cost of quality. In many practical cases, though, financial incentives will be necessary to ensure a sufficient number of annotators.

4.2.2 Base Interface (UI1)

UI1 implements all requirements described in Section 4.2.1 and forms the basis for the other UIs in this research. Figure 4.1 shows an example conversation in this UI.

The annotation process is preceded by a training phase. The training UI is implemented as an overlay shading any element not included in the training phase. A brief text describes the conversations and their context, introduces the annotation task itself and provides an estimate of total duration. Then, both main elements of the interface are introduced. The conversation is explained

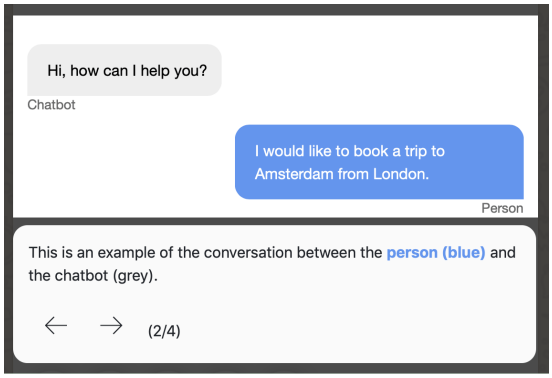


Figure 4.2: Training step to introduce the conversations, highlighting the interface element in detail and darkening all other visible elements.

first, see Figure 4.2, followed by an explanation on rating user satisfaction. The training ends with a highlight of the 'save' button. The user is then presented the main annotation interface.

In the main annotation interface, the tasks of reading the conversation and providing a rating are captured in two UI elements. Task descriptions are presented using a bold font, contrasting colour and numbers indicating order. UI elements for tasks are positioned in vertical order, a configuration known as 'cascaded'. Cascading encourages annotators to finish the one tasks before starting on the next but scrolling is required for long conversations.

UI elements for the 'reading' task were designed to be familiar to annotators. The UI used the convention to position messages from the chatbot to the right and messages from the chatbot to the right as in e.g. Facebook Messenger, Whatsapp and Signal. It may be conventional to flip this positioning in other cultural contexts. Besides positioning, colour-coding is used to differentiate between messages from chatbot and chatbot user.

The annotation task is defined as: "Rate the user satisfaction of the conversation with the chatbot". The lowest and highest value of user satisfaction are labelled "Very Satisfied" and "Very Dissatisfied", respectively. Intermediate ratings are not labelled. A rating is selected by clicking one of the five star-shaped buttons.

A sizeable green button labelled "save your results" appears after rating a conversation.

4.2.3 Interface with Definitions (UI2)

UI2 was developed to answer RQ1 from Section 4.1. In order to do so, definitions for what constitutes user satisfaction and definitions for points on the Likert scale were added. The following definition of user satisfaction was presented: "The experience of the user's goals or desires being fulfilled by the chatbot".

Definitions of points on the Likert scale were taken from [369]: "Very dis-

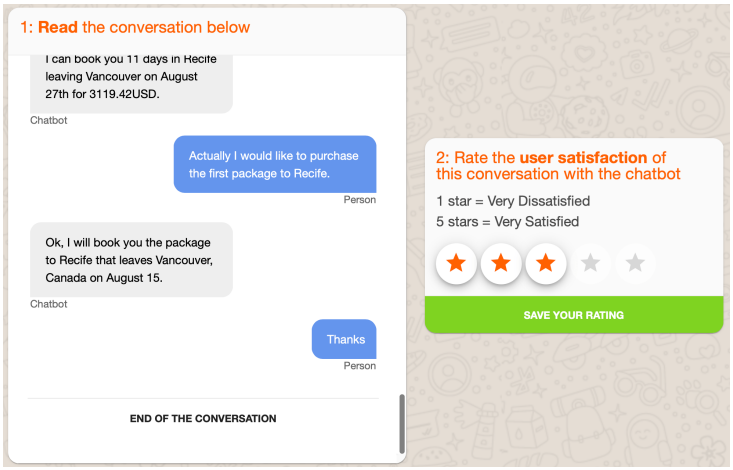


Figure 4.3: UI3: variation focusing on the separation between tasks and complexity reduction.

satisfied", "Dissatisfied", "Unsure", "Satisfied", and "Very Satisfied", .i.e. they express which levels is associated with users actually being satisfied. We hypothesize that the inclusion of definitions in UI2 increases understanding of the task when compared to UI1.

4.2.4 Interface with Separated Tasks (UI3)

UI3 was designed to answer RQ2 and is presented in Figure 4.3. The cascading of UI elements for reading and rating from UI1 was replaced by a more clear separation between UI elements that reflect these tasks.

Both elements are ordered left to right and always visible.

This allows annotators to understand which tasks to perform at a glance. The fixed positioning of the interface elements also removes some of the scrolling required for long conversations, thus further reducing complexity. We hypothesize that a simple, i.e. non-cascading, interface increases understanding of the task when compared to UI1.

4.3 Experimental setup

An online user experiment was performed in order to measure the effectiveness of the UI variations. Participants (n=27) were recruited from a set of experts (working on dialogue systems) and university students on a voluntary basis. Participants were filtered to have a sufficient proficiency in English (self-reported). Each participant was assigned a UI variation in a round-robin scheme and tasked to rate conversations from the Maluuba 'Frames' data set [13]. This data set consists of multi-turn single-domain task-oriented conversa-

tions collected in a Wizard-of-Oz scheme and contains user-reported satisfaction ratings on a five-level scale. For this experiment, ten conversations were randomly sampled for each level, e.g. each participant in this experiment rated 50 conversations.

Data additional to the user satisfaction ratings were collected. A screen capture tool[1] was used to log interface usage to ensure that participants understood the interface and did not rush through the task. Participants were also asked to complete a brief questionnaire after rating. This questionnaire contained a request for general remarks and two directed questions: (1) whether the participant understood the task and (2) whether they understood the UI.

In order to compare the quality of annotations obtained using the different UIs, measures of inter-rater agreement (IRA) were used. Out of all available measures of inter-rater agreement, Cohen's Kappa, Fleiss' K and Krippendorff's α are most found in literature [65, 106, 144]. Krippendorff's α is most appropriate for scales of higher than nominal order, such as Likert-scale values [406], and therefore used to compare UIs 1-3. Although these measures are not directly comparable, they all provide scores from 0 to 1 where 0 represents no agreement and 1 represents perfect agreement. Therefore, a comparison with IRA ratings collected in related work is provided as well. We again stress, however, that IRA ratings computed with different measures and collected on different datasets *not directly comparable* and should be treated as a weak indication of results at best.

We use bootstrapping to establish 95% confidence intervals [144] where possible. Annotation duration was extracted from screen capture logs. These logs were only incidentally inspected otherwise, particularly in one case where an annotator failed to recognize that scrolling down would reveal additional messages in UI2. The results from this participants were removed before analysis yielding a total number of annotators of 26.

4.4 Results

Figure 4.4 shows the 95%-confidence intervals for IRA for all proposed UIs as measured by Krippendorff's α. The three user interfaces all produce a high IRA with confidence intervals starting at 0.73 for UI2 with UI3 yielding the highest IRA of 0.76. We find no significant differences in IRA and rating distribution between UIs in a two-sample z-test.

We proceed by comparing these IRA values with IRA values in related work. Table 4.1 displays IRA values, number of annotators and measures (Krippendorff's α or Cohen's κ) used from this research and related work. We strongly caution against making a direct comparison between these values, since measures, numbers of annotators and datasets differ. Considering the interpretation of these measures, though, this comparison does hint in the direction that the introduced UIs provide high quality ratings.

[1] `https://www.hotjar.com`

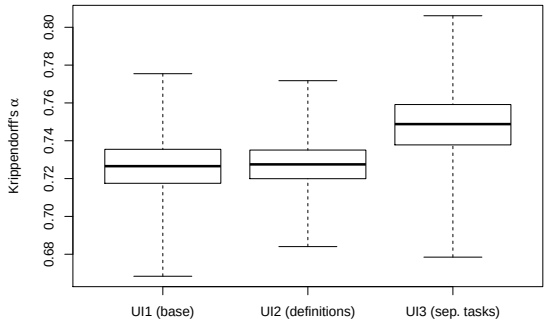


Figure 4.4: 95% Confidence Interval of the reliability scores

Table 4.1: Inter-rater agreement scores of the three UI's and similar related work

	UI1	UI2	UI3	[399]	[7]	[312]
IRA	0.74	0.73	**0.76**	<0.3	0.68	0.31
measure	K's α	K's α	K's α	C's κ	C's κ	C's κ
# annotators	9	9	8	2	2	2

All participants but one completed the questionnaire after evaluating all conversations. Seven participants were explicitly positive and found that there was little room for error. Four participants from the UI1 and one participant from UI2 mentioned they had issues in understanding the concept of user satisfaction clearly and consistently. Here, the distance between the values was mentioned and lack of context was mentioned in particular. One participant from UI3 suggested providing additional examples of satisfactory and unsatisfactory conversations in the training. In general, all tools received feedback on the duration of the task to be too long.

4.5 Discussion

In this work, we have presented three UI's for gathering high quality annotations for conversational data. We have gathered requirements from related work and implemented these in an interface (UI1). We have implemented a second interface that communicates the definition of "user satisfaction" to investigate their effect (UI2). A third interface visually separates the tasks of reading the conversation and rating it (UI3). All interfaces yield high quality annotations as measured by Inter-Rater Agreement (IRA) and received positive feedback from annotation participants in a questionnaire. These UIs are suitable for the collection of user satisfaction estimates for conversations when using third-party annotators.

In a comparison between UIs with and without a definition for 'user satisfaction', we find that the inclusion of definitions does not significantly affect annotation quality as measured by IRA. This is surprising, as the inclusion of

clear definitions can be expected to reduce the ambiguity inherent in the task. This does not mean that such definitions are not useful. Third-party annotations may be gathered to complement ratings from users [367]. In such cases, definitions may aid in ensuring that particular aspects that system designers are dissatisfied with but are not reflected in user ratings are captured in the third-party ratings.

We investigated whether a UI with cascading or a UI with separated elements for the reading and annotation tasks is to be preferred. A comparison between IRA scores indicates that a separation may be beneficial, but we find no significant effect. We hypothesize that the slight increase in IRA may be the result of the annotator being able to switch between both tasks easily. It may be necessary to reread parts of the conversation before providing an annotation. Similarly, the user is reminded of the annotation task, which may direct their reading. Similarly, the rater is able to provide an initial rating halfway through the conversation and fine-tune it while reading on in UI3. However, we also found that one user failed to recognize that scrolling was required to read the entire conversation. This may become a more pressing problem as conversations with systems become longer. Such cases can be avoided when using a cascading interface. An alternative to ensuring that annotators view the entire conversation is to disable rating input if parts of the conversation have not been scrolled down to.

A possible limitation for this work is the domain that conversations were taken from. This domain is easy to understand by annotators. Annotations in other domains may yield lower IRA. However, this will be the case for any UI to collect annotations. Quality of ratings in complex domains may benefit from more training and/or more stringent filtering of participants. Therefore, we recommend that future work carefully considers the effects of the earlier discussed requirements for their specific case. In this work, we focused on the collection of third-party annotations in general. We see the collection of third-party annotations as a suitable proxy for user ratings as a suitable follow-up from this work. Finally, we believe that minor improvements are possible, such as allowing users to return the examples from training at any time. Future work will have to address this as well.

5

Strategic Workforce Planning with Deep Reinforcement Learning

This chapter presents a simulation-optimization approach to strategic workforce planning based on deep reinforcement learning. A domain expert expresses the organization's high-level, strategic workforce goals over the workforce composition. A policy that optimizes these goals is then learned in a simulation-optimization loop. Any suitable simulator can be used, and we describe how a simulator can be derived from historical data. The optimizer is driven by deep reinforcement learning and directly optimizes for the high-level strategic goals as a result. We compare the proposed approach with a linear programming-based approach on two types of workforce goals. The first type of goal, consisting of a target workforce, is relatively easy to optimize for but hard to specify in practice and is called *operational* in this work. The second, *strategic*, type of goal is a possibly non-linear combination of high-level workforce metrics. These goals can easily be specified by domain experts but may be hard to optimize for with existing approaches. The proposed approach performs significantly better on the strategic goal while performing comparably on the operational goal for both a synthetic and a real-world organization. Our novel approach based on deep reinforcement learning and simulation-optimization has a large potential for impact in the workforce planning domain. It directly optimizes for an organization's workforce goals that may be non-linear in the workforce composition and composed of arbitrary workforce composition metrics.

Based on [P6]:
Yannick Smit, Floris den Hengst, Sandjai Bhulai and Ehsan Mehdad
Strategic Workforce Planning with Deep Reinforcement Learning
International Conference on Machine Learning, Optimization, and Data Science 2022

5.1 Introduction

In order to achieve their strategic goals, organizations need to have the right people in the right place at the right time. *Strategic workforce planning* (SWP) is the business process in which the required actions to meet an organization's workforce needs are identified [11]. SWP has been recognized as an important problem across sectors [31, 45, 66] and is expected to grow in importance with knowledge and human capital becoming increasingly important drivers of economic growth [300]. Workforce planning helps organizations with forecasting their workforce needs given a range of possible business scenarios and includes predicting the impact of various programs and policies on talent attraction and retention, showing how the impact varies across different segments of the workforce, modeling the impact of employee attrition and movements within the organization, and quantifying the financial impact of workforce decisions [11].

SWP problems are challenging since they require a deep understanding of the organization's high-level strategic goals and constraints on the one hand and technical knowledge to express these as an optimization problem on the other. The problem formulation should correctly capture the organization's workforce goals and constraints into its objective, address the aforementioned aspects of uncertainty, and be both actionable and computationally tractable. As a result, achieving impact with SWP typically requires careful collaboration between experts from the HR and analytics domains.

The SWP problem has attracted substantial interest from researchers as a result. Historically, these have focused on relatively simple and specific settings, e.g., problems of a relatively small scale [288], with a homogeneous workforce [45, 324], and an objective function linear in the workforce composition [111, 131]. Recently, researchers have addressed some of these limitations with more advanced techniques that explicitly include uncertainty of the workforce dynamics [161], that include employee attributes, such as age, skill, and position [31, 72], and that use a piece-wise linear objective [73]. Although more general than previous methods, these still rely on problem specifics to cast the organizations' goals and constraints into a tractable optimization problem. This limits their applicability and comes at a significant analysis and modeling burden.

In this work, we propose a generic and widely applicable approach. In our approach, a policy that optimizes a strategic workforce objective is derived with deep reinforcement learning (DRL). Since DRL does not depend on the specifics of the objective, it can be defined as a non-linear combination of high-level workforce metrics. The optimal policy is determined with DRL in a simulation-optimization loop. The optimization step in this loop does not depend on the internals of the simulator, so that the approach can be applied to a wide range of simulators. We also describe how a simulator can be estimated from data on historical workforce compositions so that only the objective and a data set are required as inputs. Additionally, our approach is capable of handling large problems and fine-grained decision-making as a result of the usage of neural networks in estimating the optimal policy. Our approach improves the usability,

granularity, and quality of SWP decision support.

5.1.1 Related Work

The application of different simulation paradigms in finding the optimal workforce planning decisions is very popular; see [22, 167, 171] and also see [11] for a discussion of simulation in workforce planning in industry. The adoption of deep reinforcement learning for simulation-optimization has recently become popular in academia and industry; see [145], Pathmind and project Bonsai by Microsoft. To the best of our knowledge, however, no studies have proposed to address SWP with DRL, which brings various benefits to this domain: it does not require any specific domain knowledge, scales well to large problems, and makes no assumptions on, e.g., linearity of the objective function.

This work is organized as follows. We first introduce SWP as an optimization problem, including the modeling of the workforce dynamic and the formulation of optimization objectives. We then introduce the simulation-optimization loop and detail the DRL optimizer. We describe the experimental setup and results, which show that our approach finds suitable policies for high-level objectives for both a synthetic and real-life organization. We conclude that our approach enables direct optimization of strategic workforce goals.

5.2 Strategic Workforce Planning as Optimization

In this section, we present a quantitative framework for SWP. We first detail a descriptive model of the workforce. This model factors the total workforce into groups of individuals with similar attributes of interest called *cohorts*. Attributes, such as productivity, skills, and manager status, can be included based on the goals and constraints of the organization. We then detail how the dynamics are modeled. Finally, we describe how strategic workforce goals and constraints can be formulated as optimization objectives.

5.2.1 Cohort Model

We define employee attributes as a set of variables $Y = (Y_1, \ldots, Y_m)$ so that each employee with attributes $(Y_1 = y_1, \ldots, Y_m = y_m)$ can be described by values $(y_1, \ldots, y_m)$ and all employees with the same values can be grouped into the same cohort $C_i \in \mathcal{C} = \{C_1, \ldots, C_n\}$. The number of cohorts n depends on the number of attributes m and the cardinalities $|Y_i|$ of these attributes, i.e., $n = |\mathcal{C}| = \prod_{i=1}^{m} |Y_i|$. Note that n grows as a combination of attributes, so that more fine-grained modeling results in a larger number of cohorts quickly.

We now turn to a model of the evolution of a workforce over time. Specifically, we consider discrete time steps of an arbitrary fixed length (e.g., monthly, quarterly, or yearly) $0 < t \leq T$ for some finite horizon $T < \infty$. At each time point t, the number of employees for a particular cohort C_i is defined as a

random variable (r.v.) $X_{i,t} \in \mathbb{N}_{\geq 0}$ and the total workforce as a combination of all cohorts $X_t = \left(X_{(1,t)}, \ldots, X_{(n,t)}\right) \in \mathbb{N}^n_{\geq 0}$. The dynamics of these so-called *headcounts* can now be modeled as a Markov chain. Its state space consists of all possible headcount compositions. We assume a scalar $X_{\max} < \infty$ for the maximum number of employees per cohort and define the state space of the Markov chain $\mathcal{S} = \{s \in \mathbb{N}^n_{\geq 0} | s \leq (X_{\max})^n\}$.

For any organization and for any time step, we know that an individual can either (i) leave the organization organically due to, e.g., retirement, voluntarily leaving etc. (ii) leave the organization as a result of a management decision, (iii) move from one cohort to another cohort organically, (iv) be moved from one cohort to another cohort by the organization and (v) enter the organization. With this knowledge, the transition function can be factorized into components, so that for every t:

$$X_{t+1} = X_t - O_t - L_t + \mathbb{1}^n M_t - \mathbb{1}^n M'_t + \mathbb{1}^n N_t - \mathbb{1}^n N'_t + H_t, \tag{5.1}$$

where $\mathbb{1}^n$ is an n-dimensional vector of ones and (i) O_t an n-dimensional r.v. representing organic leavers per cohort, (ii) L_t an n-dimensional r.v. representing organization-initiated leavers, (iii) M_t an $n \times n$ random matrix of employees moving between cohorts organically, (iv) N_t an $n \times n$ random matrix of moves between cohorts initiated by the organization, and (v) H_t an n-dimensional r.v. of new hires. This model describes how the workforce changes over time and it allows to easily formalize strategic workforce goals as optimization objectives as described in the next section.

5.2.2 Optimizing the Cohort Model

In this section, we cast the SWP problem as an optimization problem. The first step is to identify the actions available to the organization. We assume that these are direct and indirect controls on the Markov chain in Equation 5.1. In general, the transitions L_t, N_t, and H_t are controlled by the organization directly. Additional controls may be in place to affect the other r.v.'s indirectly. For example, an employee retention plan can be included to affect the attrition O_t. The cohort model supports both direct and indirect controls, and these can be included based on the organization's needs.

The organization should take those actions that result in the most suitable workforce at every time step. We here formalize the organization's actions as some set $\mathcal{A}$ and a particular action at time t as $A_t \in \mathcal{A}$ and refer the reader to Section 5.4 for examples. We assume that each workforce composition X_t and A_t can be assigned a numerical value corresponding to the particular SWP goal of the organization with some function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. The objective, now, is to maximize this value over time by sampling appropriate states and

actions in the system in Equation 5.1 until some horizon T:

$$\begin{aligned} A^* &= \arg\max_{A_0,\dots,A_T} \mathbb{E}\left[\sum_{t=0}^{T} r(X_{t+1}, A_t)\right] \\ \text{s.t. } X_{t+1} &= X_t - O_t - L_t + \mathbb{1}^n M_t - \mathbb{1}^n M'_t + \mathbb{1}^n N_t - \mathbb{1}^n N'_t + H_t, \\ &\text{and } O_t, L_t, M_t, N_t, H_t \text{ dependent on } A_t \\ &\text{for } t = 0, \dots, T, \text{ and a given } X_0. \end{aligned} \tag{5.2}$$

Having defined the general optimization objective, we now turn to examples of suitable reward functions. A reward function should reflect the strategic workforce goals of the organization accurately. Because of the strategic nature of SWP, a goal is usually composed of multiple terms. General terms such as headcount and budget, SWP-specific terms such as average span of control[1], job level[2] and manager status, and finally, organization-specific metrics such as productivity, skills, and diversity may all be included.

5

Strategic Workforce Goals

The example strategic workforce goal is composed of three components, here presented by decreasing importance. The primary component consists of bounds for headcounts for each cohort. The second component contains a target average span of control across the organization. In general, such a target span of control is attained by multiple workforce compositions. The third component, therefore, specifies that minimal salary costs are preferred. We formalize this strategic goal by formalizing each component and then combining the components in an overall objective.

To formalize the objective based on headcount bounds, we penalize cohorts that are out of bounds:

$$r_{\mathrm{b}}(X_t) := -\sum_{i=1}^{n} \mathbb{1}_{\{X_{i,t} \notin [\ell_i, u_i]\}}, \tag{5.3}$$

where $\ell, u \in \mathbb{N}^n_{\geq 0}$ are lower and upper bounds for all n cohorts. Next, we define a component for achieving the target span of control. It is similar to the objective for target headcounts in Equation (5.9):

$$r_{\mathrm{soc}}(X_t) := \exp\left(\frac{-\alpha_{\mathrm{soc}}\left(\mathrm{soc}(X_t) - G_{\mathrm{soc}}\right)^2}{G^2_{\mathrm{soc}}}\right), \tag{5.4}$$

where $G_{\mathrm{soc}} > 0$ is a target average span of control, $\alpha_{\mathrm{soc}} > 0$ a precision parameter, and $\mathrm{soc}(X_t)$ a function that returns the average span of control for X_t:

$$\mathrm{soc}(X_t) := \frac{X_{(n/2+1,t)} + \cdots + X_{(n,t)}}{X_{(1,t)} + \cdots + X_{(n/2,t)}}. \tag{5.5}$$

[1] The average number of direct reports of managers in the organization.

[2] A metric to express responsibilities and expectations of a role in the organization, usually associated with compensation in some way.

The third and final component can be formalized based on a function $\text{sal}(X_t)$ that returns the estimated total salary cost for a workforce X_t. This final component has the lowest priority. Therefore, we only assign a positive value based on salary if the span of control component is sufficient, as expressed by a lower bound $\ell_{\text{soc}} \in [0, 1]$:

$$r_{\text{sal}}(X_t) := \begin{cases} r'_{\text{sal}}(X_t), & \text{if } r_{\text{soc}}(X_t) > \ell_{\text{soc}}, \\ 0, & \text{otherwise}, \end{cases} \tag{5.6}$$

for a salary normalized to $[0, 1]$ based on the cohort bounds ℓ, u:

$$r'_{\text{sal}}(X_t) := \text{clip}\left(\frac{\text{sal}(X_t) + \text{sal}(\ell)}{\text{sal}(\ell) - \text{sal}(u)}, 0, 1\right). \tag{5.7}$$

The *strategic* objective is composed of the sub goals in Equations (5.3)-(5.6). We combine the components to reflect all sub goals states earlier:

$$r_{\text{s}}(X_t) := r_{\text{b}}(X_t) + r_{\text{soc}}(X_t) + r_{\text{sal}}(X_t). \tag{5.8}$$

The simulation-optimization approach proposed in this work targets the direct optimization of objectives that reflect an organization's *strategic* workforce goals and that may be non-linear and composed of arbitrary workforce metrics.

Operational Workforce Goals

Another type of workforce goal is to meet a particular known demand for employees in each cohort. This type of goal is relatively easy to optimize for but hard to specify in practice. For this goal, a reward can be assigned based on a distance between the current workforce X_t and the known target composition $X^* = (X_1^*, \ldots, X_n^*)$ for all n cohorts. To ensure that the cohorts contribute uniformly to this reward, headcounts need to be scaled to $[0, 1]$. Now, the following rewards an observed headcount $X_{i,t}$ for a single cohort i based on its target X_i^*:

$$r_{\text{c}}(X_{i,t}) := \begin{cases} \exp\left(\frac{-\alpha(X_{i,t} - X_i^*)^2}{(X_i^*)^2}\right), & \text{if } X_i^* > 0, \\ \exp\left(-\alpha X_{i,t}^2\right), & \text{if } X_i^* = 0, \end{cases} \tag{5.9}$$

where the so-called *precision* parameter $\alpha > 0$ specifies how strictly to penalize sub-optimal headcounts. A simple operational reward averages over all n cohorts:

$$r_{\text{o}}(X_t) := \frac{1}{n} \sum_{i=1}^{n} r_{\text{c}}(X_{i,t}), \tag{5.10}$$

These *operational* workforce goals are generally easy to optimize for using established optimization techniques since they can be cast as linear optimization problems. However, defining the required headcounts for all cohorts to meet high-level workforce goals is very hard in practice.

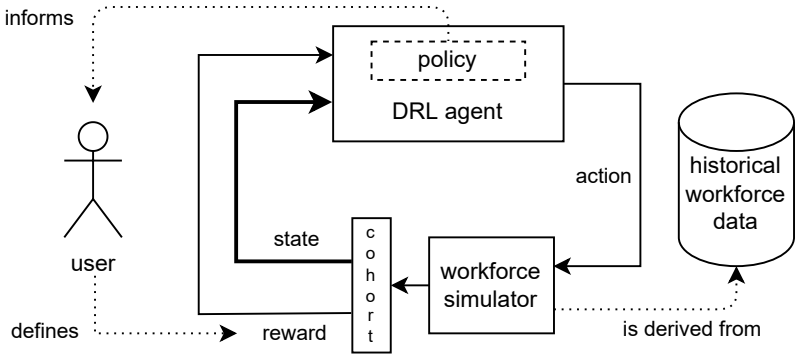


Figure 5.1: Overview of the simulation-optimization approach. A user specifies the organization's strategic workforce goal. A black-box workforce simulator is then used to find a policy that directly optimizes for the goal with DRL. This policy helps the user making informed workforce decisions.

5.3 Simulation-Optimization with Deep Reinforcement Learning

We propose a simulation-optimization loop for solving SWP problems. Figure 5.1 contains a visualization of this loop. First, the user specifies the strategic workforce goals of the organization as a reward function to maximize. This function may be any arbitrary, e.g., a non-linear function defined over a cohort representation of the workforce. Next, a policy is learned by a DRL agent by interacting with a simulator. This simulator can be any suitable black-box simulator that outputs a cohort representation of the workforce and can take into account the decisions made by the agent. By using DRL for optimization, the strategic goals are optimized for directly, and, hence, the resulting policy informs the user in taking the right workforce decisions for their strategic workforce goals. If historical data of the workforce is available, then this simulator can be learned from data as described in Section 5.3.2.

5.3.1 Deep Reinforcement Learning for Workforce Planning

Formally, we cast the SWP problem as a Markov decision process (MDP) $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma \rangle$, where $\mathcal{S}$ is a state space, $\mathcal{A}$ an action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ a transition function, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ a reward function, and $\gamma \in (0, 1]$ a discount factor to balance immediate and future rewards. The decisions of the agent are defined by its policy $\pi_\theta : \mathcal{S} \times \mathcal{A} \to [0, 1]$, which depends on a parameter vector θ which can, e.g., be a neural network. The goal of the agent is to maximize the expected discounted return $J(\theta) := \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T-1} \gamma^t r_{t+1}\right]$, which can be done by tuning parameters θ with an algorithm that alternates simulating experience

in the environment and optimizing the policy. Here $r_{t+1} = r(s_t, a_t)$ and $\mathbb{E}_{\pi_\theta}$ indicates that $s_{t+1} \sim \mathcal{P}(\cdot, a_t, s_t)$ and $a_t \sim \pi_\theta(\cdot|s_t)$.

In the proposed framework, the state space of the MDP is equal to the state space of the Markov chain over headcounts, i.e., $\mathcal{S} = \{s \in \mathbb{N}_{\geq 0}^n | s \leq (X_{\max})^n\}$. The optimization algorithm uses a neural network to evaluate the value of each state. To help the convergence of the network and significantly reduce training time, the inputs to the network are normalized. Hence, we implement the state space of the MDP as a continuous space $\hat{\mathcal{S}} = [0, 1]^n$, where states are defined as $s_t = \left(\frac{X_{1,t}}{X_1^{max}}, \ldots, \frac{X_{n,t}}{X_n^{max}}\right)$ for training. The action space is given by the controls over the workforce as described in Section 5.2, for example, a multi-discrete set of numbers of employees that enter or leave the organization for each cohort. For the purposes of optimization, the dynamic model $\mathcal{P}$ is assumed to be unknown so that any suitable simulator can be used. The reward function is defined by an end user based on the organization's strategic workforce goals. It can be composed of arbitrary and non-linear workforce metrics of interest to the organization, see Section 5.2.2 for details and examples.

In the optimization step a policy is updated to optimize the given objective. This update is performed by approximate gradient ascent on θ, i.e., iteratively update $\theta_{k+1} = \theta_k + \eta\widehat{\nabla_\theta J(\theta)}$. The gradient $\nabla_\theta J(\theta)$ is estimated by $\hat{\mathbb{E}}_t \left[\nabla_\theta \log \pi_\theta(a_t|s_t)\hat{A}_t\right]$, where $\hat{\mathbb{E}}_t$ denotes an empirical estimate over a batch of samples collected over time and $\hat{A}_t$ is an estimator of the advantage function. While our approach is generic to various optimization algorithms, we propose to use Proximal Policy Optimization (PPO) as it has shown to be suitable in high-dimensional settings with non-linear rewards [313].

5.3.2 Simulating the Workforce

This section details how the dynamics of a cohort model from Section 5.2.1 can be estimated from data. Estimation is necessary for two reasons. Firstly, the dynamics may simply not be available to the organization. Secondly, it may be problematic to fully elaborate the dynamics up-front due to the complexity of the problem. Specifically, the size of the state space of the cohort model Markov chain grows exponentially in the number of cohorts. As a result, it becomes infeasible to analytically define it fully for reasonably large organizations.[3] Hence, we estimate the dynamics from data with simplifications that apply to the cohort model.

In many cases, Equation (5.1) can be simplified by assuming limited control of the workforce by management. For example, if we only model management-controlled hires and leavers, N_t becomes equal to the zero matrix and $A_t := H_t - L_t$ for the combined movement of hires and leavers by the organization. The part of the transition function that is out of management control is now given by $X_{t+1} = X_t - O_t + \mathbb{1}M_t - \mathbb{1}M'_t$. Note that the diagonal entries of

[3]For a model with $n = 30$ cohorts and $X_{\max} = 100$ maximum employees per cohort, the number of transitions in the Markov chain is $|\mathcal{S} \times \mathcal{S}| = \prod_{i=1}^{n}(S_{\max} + 1)^2 \approx 10^{120}$.

M_t can be chosen arbitrary, since $(M_t - M'_t)_{i,i} = 0$ for all $i = 1, \ldots, n$. By realizing that the numbers of employees that remain in cohort i is equal to the headcount of cohort i minus the number of employees that move to any other cohort or organically leave the organization, we may set $M_{i,i,t} = X_{i,t} - \sum_{j=1, j\neq i}^{n} M'_{i,j,t} - O_{i,t}$, or $X_t = \mathbb{1}M'_t + O_t$. It follows that we can then simply write $X_{t+1} = \mathbb{1}M_t$ for the stochastic dynamics of the cohort model in general.

We observe that all employees within a certain cohort are indistinguishable for the purposes of SWP. Hence, approximation of the dynamics of Equation (5.1) at cohort level is sufficient for the purposes of this work. We, therefore, model the movement of employees between cohorts based on the attributes that describe the cohorts. We define the transition probability matrix $P(t) \in [0,1]^{n\times n}$ by letting $p_{i,j}(t)$ be the probability that an employee moves from cohort i at time t to cohort j at time $t+1$. We additionally assume time-homogeneous transition probabilities, i.e., $P(t) \equiv P$. Under these assumptions, the rows of the random matrix M_t follow a multinomial distribution, i.e., for $i = 1, \ldots, n$, $(M_{i,1,t}, \ldots, M_{i,n,t}) \sim \text{Mult}\,(X_{i,t}, P_i)$, where P_i denotes the i-th row of P.

The transition probability matrix P can be estimated from data that takes record of the cohorts of individual employees over a time period $t = 1, \ldots, T$. Let $m_{i,j,t}$ denote the number of employees that are in cohort i at time $t-1$ and in cohort j at time t. Then the maximum likelihood estimator of $p_{i,j}$ is

$$\hat{p}_{i,j} = \frac{\sum_{t=1}^{T} m_{i,j,t}}{\sum_{t=1}^{T} X_{i,t}}. \tag{5.11}$$

For any time step t and action A_t, the dynamics of the workforce over time can now be simulated by sampling the movement matrix M_t from the multinomial distribution described above and computing

$$X_{t+1} = \mathbb{1}^n M_t + A_t. \tag{5.12}$$

5.4 Experimental Setup

This section details the experimental setup, which was designed to answer the following research questions:

1. How does the proposed simulation-optimization approach perform,
   (a) on an operational workforce objective?
   (b) on a strategic workforce objective?
   (c) for a varying employee mobility?
2. Are firing constraints best implemented with a masked policy or an updated objective (penalty for illegal fires)?

We compare the results on a baseline based on linear programming (LP) proposed recently [72]. We evaluate these approaches in two cases. The first is a synthetic organization, and the second is a real-life use case from an international bank to validate the results in practice. We first describe the overall setup, then the baseline, detail the organizations, and include implementation details.[4]

To investigate research question 1a and 1b, we train a reinforcement learning agent for both the operational and strategic tasks in Equation (5.9) and Equation (5.8). We evaluate both the trained agent and the heuristic baseline described in Section 5.4.1 and compare the performance based on the average reward metric, in the manner as described in Section 5.4.2.

5.4.1 Baseline

We devise a baseline based on linear programming to compare the performance of the proposed simulation-optimization approach. This baseline was proposed in [72] and makes a number of additional assumptions that allow for efficient solving of the SWP problem. We describe this baseline in detail in this section.

Due to the size of the state space of the Markov chain that describes the workforce dynamics, this stochastic model cannot be used directly with a linear solver. Therefore, we consider a deterministic approximation of Equation (5.12), by replacing the random variables involved with their expectation. This operation, known as mean-field approximation, is justified for large-scale organizations as a result of the functional law of large numbers; see, e.g., [72]. For Equation (5.12) we obtain

$$X_{t+1} \approx \mathbb{E}\left[\mathbb{1}M_t + A_t\right] = X_t P_{\cdot,i} + A_t, \tag{5.13}$$

where $P_{\cdot,i}$ denotes the i-th column of the transition probability matrix P. Additionally, we optimize for one time step at a time instead of the whole trajectory $t = 0, \ldots, T$. This is reasonable when the rewards do not depend on time and are given at each time step. In that case, there are no situations where it is required to sacrifice short-term gains for long-term profit.

Consider the target level reward defined in Equation (5.9) and assume for simplicity that $X_i^* > 0$ for all $i = 1, \ldots, n$. Under the aforementioned assumptions, this version of the SWP problem is given by: find

$$A_t^* = \arg\max_{A_t} \frac{1}{n} \sum_{i=1}^{n} \exp\left(\frac{-\alpha(X_{i,t+1} - X_i^*)^2}{(X_i^*)^2}\right), \tag{5.14}$$

such that $X_{i,t+1} = \sum_{j=1}^{n} p_{ji} X_{j,t} + A_{i,t}$ for $t = 1, \ldots, T$. Substituting the latter expression in the former, and by noting that each term of the sum is maximized when the term in the exponential is equal to zero, we see that $A_{i,t}^* = X_i^* - \sum_{j=1}^{n} p_{ji} X_{j,t}$. The optimal continuous actions are then mapped

[4] Code and data for hypothetical use case available at `https://github.com/ysmit933/swp-with-drl-release`. Real-life use case data will be made available upon request.

to the discrete set of possible hiring options $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$. Hence, the decision rule becomes

$$A^*_{i,t} = \Pi_{\mathcal{A}_i}\left(X^*_i - \sum_{j=1}^{n} p_{ji} X_{j,t}\right), \tag{5.15}$$

where $\Pi_{\mathcal{A}_i}(a) := \arg\min_{a' \in \mathcal{A}_i} |a - a'|$.

To develop a heuristic for the combined reward function, we make the same assumptions as for the target level heuristic. We then consider the state that yields the highest immediate reward, given by $X^* = \arg\max_{x \in \mathcal{S}} r(x)$. We use this as a target level to aim for by applying the target level heuristic Equation (5.15).

5.4.2 Training Setup

The reinforcement learning agent is trained for a maximum number of training steps T^{max} specified by the user. At the start of each episode, a random starting state in the neighborhood of X_0 is generated to ensure sufficient exploration of all relevant parts of the state space. This is done by uniformly sampling a state from the interval $[(1-\beta)X_i(0)/X_i^{max}, (1+\beta)X_i, 0/X_i^{max}]$, where $\beta \in [0,1]$ determines the random spread across the state space. The episode ends after T time steps, at which point the environment resets to a new random starting state. After each T^{eval} number of time steps, the agent is evaluated on an evaluation environment, which is identical to the training environment except for a deterministic start at X_0 and the best performing agent is stored.

When the training process has terminated, we test the trained model on the evaluation environment with a fixed starting state (as a default for 1,000 episodes) and collect several metrics to assess the quality of the model. In particular, during an episode of T time steps, we collect the average reward $\frac{1}{T}\sum_{t=1}^{T} r(X_t)$ and the number of constraint violations $\sum_{t=1}^{T}\sum_{i=1}^{n} \mathbb{1}_{\{A_i(t) \text{ is illegal}\}}$.

5.4.3 Hypothetical Organization

For the hypothetical organization, we consider a model with four cohorts, labeled by M1, M2, C1, and C2 (two cohorts of managers and two cohorts of contributors). We suppose the probability transition matrix is given by

$$P = \begin{pmatrix} 0.98 & 0 & 0 & 0 \\ 0.01 & 0.93 & 0 & 0 \\ 0 & 0.04 & 0.92 & 0.005 \\ 0 & 0.01 & 0.01 & 0.96 \end{pmatrix},$$

and we let $X_0 = (20, 50, 100, 300)$ be the starting state. The hiring options are set to $\mathcal{A}_1 = \{-2, -1, 0, 1, 2\}$, $\mathcal{A}_2 = \{-5, -1, 0, 1, 5\}$, $\mathcal{A}_3 = \{-10, -2, 0, 2, 10\}$, and $\mathcal{A}_4 = \{-25, -5, 0, 5, 25\}$. The maximum cohort sizes are $X^{max} = 2X_0$, the

random starting state percentage is $\beta = 0.25$, the time horizon is $T = 60$, and the salary costs are set to $C^{sal} = (10000, 6000, 4000, 2000)$. The target level objective is $X^* = X_0$ (and remain at the same levels as the starting state), with a default precision of $\alpha = 10$. The combined reward parameters are given by $\ell = 0.75X_0$, $u = 1.25X_0$, $G_{soc} = 7$, $\ell_{soc} = 0.9$.

On top of the transitions introduced before, we vary the employee mobility in order to answer research question 1c. In order to answer this question, we evaluate the approach on transition matrices

$$P_\ell = \begin{pmatrix} 1-\ell & 0 & 0 & 0 \\ \ell/2 & 1-\ell & 0 & 0 \\ 0 & \ell/2 & 1-\ell & 0 \\ 0 & 0 & \ell/2 & 1-\ell \end{pmatrix},$$

for mobility rates $\ell \in \{0, 0.01, \ldots, 0.1\}$. For each of these environments, and for both the operational and strategic tasks, a reinforcement learning agent is trained and evaluated. Next, its performance, based on the average reward obtained, is compared to the heuristic baseline.

5.4.4 Real-life use case

To investigate the performance of our solution method on a real-life use case, we use the following model based on actual headcounts in one particular department of the Bank. The 14,105 employees of this segment of the organization are divided into cohorts based on manager status (manager or contributor) and based on five distinct job levels, resulting in a cohort model consisting of $n = 10$ cohorts. We label the cohorts as Manager-1, ..., Manager-5, Contributor-1, ..., Contributor-5. The transition probabilities between these cohorts are estimated based on monthly employee data for a period of 48 months. For both tasks, starting state X_0 is set to the workforce at the beginning of the period and target state X^* to the workforce at the end of the period for the operational goal.

For the strategic goal, we use cohort bounds $\ell_i = 0.75X_i(0)$ and $u_i = 1.25X_i(0)$, and the goal for span of control is $G_{\text{soc}} = 7$, with $\ell_{\text{soc}} = 0.9$. Costs associated with salary and management initiated hires and leavers were set in collaboration with an expert in the organization. The hiring options were chosen based on the cohort sizes and include the option to hire or fire zero, a few, many, or a moderate number of employees. The maximum number of employees that could be hired or fired was roughly ten percent of the starting cohort size. For example, the hiring options for cohort Manager-1 were given by the set $\mathcal{A}_1 = \{-25, -5, -1, 0, 1, 5, 25\}$.

To investigate research question 2, we implement three methods to constrain the choices for management-initiated leavers. The first method is a masked policy, for which the illegal actions are removed from the action space by setting the corresponding action probabilities to zero. For the second method, the agent receives a large negative reward for selecting an illegal action. Finally, we constrain the agent to hires only, i.e. in which all leaving employees do so

Figure 5.2: Normalized cumulative training rewards.

Table 5.1: Average normalized cumulative rewards and 95% confidence interval for both tasks on both organizations. **Bold** denotes significant best per task ($p = 0.99$).

	Synthetic		Real-life	
	Operational	Strategic	Operational	Strategic
LP	$\mathbf{0.98 \pm 0.030}$	0.41 ± 0.374	$\mathbf{0.99 \pm 0.010}$	0.12 ± 0.106
SO (ours)	0.94 ± 0.033	$\mathbf{0.83 \pm 0.213}$	0.92 ± 0.015	$\mathbf{0.98 \pm 0.026}$

organically. We then train reinforcement learning agents for both the operational and strategic tasks and compare the performance of the unconstrained agent, the masked agent, the penalty-receiving agent, the no-fire agent, and the baseline heuristic.

5.5 Results

In this section, we look at all results associated with research questions 1a-2 presented in the previous section. We first look at the convergence of the proposed approach in Figure 5.2 and find that the proposed SO approach con-

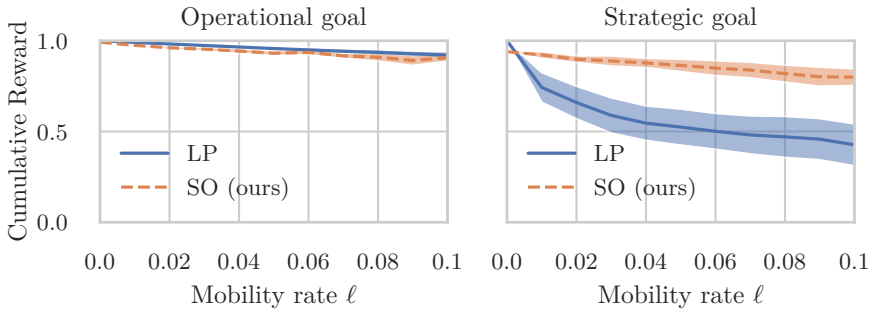


Figure 5.3: Normalized cumulative rewards for varying mobility rates.

Table 5.2: Average normalized cumulative rewards and constraint violations (% of total decisions), with 95% confidence intervals. **Bold** denotes significant best ($p = 0.99$). LP=linear programming, U=unconstrained, M=masked, P=penalty and OH=only hires.

	Operational		Strategic	
	Reward	# Violations (%)	Reward	# Violations (%)
LP	$\mathbf{0.99 \pm 0.010}$	16.83 ± 3.05	0.12 ± 0.106	13.84 ± 2.96
U	0.92 ± 0.015	6.94 ± 1.56	$\mathbf{0.98 \pm 0.026}$	21.72 ± 4.39
M	0.74 ± 0.030	$\mathbf{0.00 \pm 0.00}$	0.75 ± 0.135	$\mathbf{0.00 \pm 0.00}$
P	0.87 ± 0.046	0.34 ± 0.07	0.74 ± 0.060	$\mathbf{0.00 \pm 0.00}$
OH	0.79 ± 0.041	$\mathbf{0.00 \pm 0.00}$	0.94 ± 0.061	$\mathbf{0.00 \pm 0.00}$

verges quickly. Next, we compare the resulting policies with an LP baseline on a test set. Table 5.1 shows that the proposed approach performs close to the optimum of the LP baseline on the operational objective and significantly outperforms the baseline on the strategic objective. We move on to research question 1c by looking at the effect of increasing the employee mobility in Figure 5.3. It shows that the the proposed SO approach is robust against a wide range of mobility levels and that its benefits increase with increasing workforce mobility. The proposed approach shows to be more robust to the stochastic nature of SWP for this nonlinear optimization objective than the LP baseline.

Finally, we compare our approach in a setting with constraints on the organization's control of leavers in Tables 5.2. Here, we find that we can effectively take the organization's constraints into account using either masking, with a negative reward (penalty) or by only including hires in the action space. Out of these, the 'only hires' variant yields the best results with respect to reward and constraint adherence, with rewards close to its unconstrained counterparts without any constraint violations.

5.6 Discussion

In this work, we have presented a simulation-optimization approach to strategic workforce planning. The approach optimizes workforce decisions with DRL by interacting with a simulator. Any suitable simulator can be used because the optimization step does not depend on its internals. We propose to use a Markov chain simulator *learned* from historical data. By doing so, the full loop only requires a data set of historical workforce compositions and the organization's objective as inputs. These objectives may be composed of arbitrary workforce metrics of interest that may be non-linear in the workforce composition. The approach optimizes these objectives *directly*, so that the resulting policy can easily be used to ensure a high impact of the SWP efforts.

We have evaluated the proposed approach on a synthetic and a real-world organization and found that it converges quickly. More so, we compared the quality of the obtained policy to a baseline from the literature. In this com-

parison, we first targeted an objective composed of workforce metrics. Such objectives are easy to define and accurately reflect the organization's strategic goals. We found that our approach significantly outperforms the baseline on this *strategic* objective and that the difference grows as mobility of the workforce increases. We secondly targeted an *operational* goal, in which the optimal workforce composition is known up-front. Such goals are easy to optimize for with established optimization approaches but hard to define in practice. Our approach performed close to the baseline in this setting. We additionally showed how the approach can take into account realistic constraints by limiting the ability of the organization to control leavers in the organization and found that removing the ability to do so has a very limited impact on overall performance.

We have shown that the proposed simulation-optimization approach is suitable for SWP. Additionally, it opens up various avenues for future work. Firstly, the approach is capable of optimizing for strategic objectives composed of arbitrary workforce metrics. It would be interesting to extend the approach with multi-objective reinforcement learning in order to compute a set of Pareto optimal policies [299]. This will increase the organization's understanding of the trade-offs involved and allow them to fine-tune their strategy. Secondly, the approach currently finds a policy that is optimal on average. While this is suitable for many use-cases, there may be some organizations that prefer a probabilistic guarantee on the minimum number of employees to, e.g., meet service level agreements. Here, risk-sensitive DRL can be employed instead of regular DRL [95]. Additionally, organizational constraints can be formalized and used within approaches that guarantee safety of the resulting policy (see Chapter 7). We believe that, with the proposed approach, these challenging and interesting research directions that will further increase the impact of SWP have become feasible in practice.

Part II

Subsymbolic RL and Symbolic Knowledge

6

Guideline-informed reinforcement learning for mechanical ventilation in critical care

Reinforcement Learning (RL) has recently found many applications in the healthcare domain thanks to its natural fit to clinical decision-making and ability to learn optimal decisions from observational data. A key challenge in adopting RL-based solution in clinical practice, however, is the inclusion of existing knowledge in learning a suitable solution. Existing knowledge from e.g. medical guidelines may improve the safety of solutions, produce a better balance between short- and long-term outcomes for patients and increase trust and adoption by clinicians. We present a framework for including knowledge available from medical guidelines in RL. The framework includes components for enforcing safety constraints and an approach that alters the learning signal to better balance short- and long-term outcomes based on these guidelines. We evaluate the framework by extending an existing RL-based mechanical ventilation (MV) approach with clinically established ventilation guidelines. Results from off-policy policy evaluation indicate that our approach has the potential to decrease 90-day mortality while ensuring lung protective ventilation. This framework provides an important stepping stone towards implementations of RL in clinical practice and opens up several avenues for further research.

Based on [P2]:
Floris den Hengst, Martijn Otten, Paul Elbers, Frank van Harmelen, Vincent François-Lavet and Mark Hoogendoorn
Guideline-informed reinforcement learning for mechanical ventilation in critical care
submitted to Artificial Intelligence in Medicine

6.1 Introduction

Reinforcement learning (RL) is a promising technique to improve decision-making in healthcare because it incorporates uncertainty into its sequential decision-making and learns from observational data. As a result, various RL solutions to inform or even automate clinical decision-making have recently been proposed [P4, 181, 266, 298]. Several challenges related to performance and safety, however, remain for putting RL into clinical practice, including the alignment of learned solutions with existing knowledge, balancing long-term and short-term objectives and avoiding negative long-term side-effects of aggressive treatment.

We set out to develop an approach that combines RL with a knowledge-driven approach to obtain a hybrid solution that benefits from the best of both worlds [372]. We propose to use existing knowledge available in treatment guidelines in order to strike the right balance between model richness and modeling effort while remaining sufficiently general to apply it to a wide range of guidelines.

We propose a framework for finding effective and guideline-compliant treatment policies by incorporating treatment guidelines into RL. The guidelines are encoded into logical constraints of two kinds. The first kind of constraint limits the available decisions whereas the second kind informs the learner of desirable properties of the patient condition via an additional reward function.

We evaluate our approach in a case study on mechanical ventilation (MV) optimization using the MIMIC-III database. In this case study, we extend an existing MV modeling approach by including a protective lung ventilation guideline designed to decrease the risk of lung injury caused by MV. We compare results obtained by clinicians in the dataset with a learned policies and include versions that copy clinicians decisions with Imitation Learning (IL), minimizes 90-day mortality with Q-Learning (QL) and include guidelines. We report results in terms of compliance to the guideline and expected 90-day mortality using multiple strategies for off-policy policy evaluation.

We find that the proposed framework produces policies that fully comply with the guideline while performing significantly better than or comparable to the clinicians in terms of mortality, depending on the selected evaluation. In a comparison between policies trained with and without knowledge of the patient condition, we find no particular benefits of these constraints. In an analysis of the differences in decision-making between clinicians and learned solutions, we find that the learned solutions select more varied actions than the clinicians. In a comparison between a guideline-compliant and a non-compliant solution, we find that the compliant solution chooses settings that are close to the noncompliant solution while avoiding extreme settings.

The proposed framework can be used to infer policies from both data and medical guidelines and assess their performance. These policies adhere to a guideline and may therefore be more trusted by clinicians. Furthermore, comparisons between guideline-compliant policies with their non-compliant coun-

terparts can shed light on the effectivenes of particular guideline statements. By combining data-driven and knowledge-driven approaches, this framework comprises an important step in closing the gap between research and practice for RL in clinical settings.

6.2 Background

RL is a framework for problems in which a sequence of decisions is to be made in an environment in order to maximize a total amount of collected reward [338]. In the medical setting, the decisions can be treatment decisions, the environment can be the patient's condition and the collected reward can be treatment effectiveness expressed as e.g. 90-day mortality. In this section, we introduce the basic notation and setting we use for RL, discuss how RL can be used to obtain policies for decision-making and how these policies can be evaluated with observational data approach called *off-policy policy evaluation* (OPE).

6.2.1 MDPs and Q-learning

The sequential decision-making problems addressed within the RL framework are formally known as Markov decision problems (MDPs). An MDP is defined as a tuple (S, A, T, R, γ) where S a set of environment states, A a set of agent actions, $T : S \times A \to \mathbb{P}(S)$ a probabilistic transition function, $R : S \times A \to [R_{min}, R_{max}]$ a reward function with $R_{min}, R_{max} \in \mathbb{R}$ and $\gamma \in [0, 1)$ a discount factor to balance immediate and future rewards. At each time step t, the agent observes the environment state (patient condition)[1] $s_t \in S$, performs some action $a_t \sim \pi \in \Pi : S \to \mathbb{P}(A)$ and collects rewards $r_t \sim R(s_t, a_t)$.

For an expectation $\mathbb{E}_\pi$ of the sum of discounted future rewards for a given policy π, values V and Q can be assigned to a state s or state-action tuple (s, a) for that π:

$$V_\pi(s) = \mathbb{E}_\pi \left[\sum_{k=t}^{T=\infty} \gamma^{k-t} r_k | s_t = s \right] \tag{6.1}$$

$$Q_\pi(s, a) = \mathbb{E}_\pi \left[\sum_{k=t}^{T=\infty} \gamma^{k-t} r_k | s_t = s, a_t = a \right]. \tag{6.2}$$

An optimal policy π^* selects actions in such a way that the highest possible discounted sum of future rewards $\sum_{k=t}^{\infty} \gamma^{k-t} r_k$ is obtained for very state $s \in S$. The optimal policy can be by estimating values $\hat{Q}_{\pi^*}$ for the optimal policy and then selecting the action according to these values deterministically:

$$\pi(a|s) = \text{argmax}_{a \in A} \hat{Q}(s, a) \tag{6.3}$$

[1] in some cases, such as within the POMDP framework, the state is composed of a history of observations, actions and rewards

or stochastically with e.g. Boltzmann action selection:

$$\pi(a|s) = \frac{e^{\hat{Q}(s,a)/T}}{\sum_{a' \in A} e^{\hat{Q}(s,a')/T}} \tag{6.4}$$

where temperature $T \in \mathbb{R}^{\geq 0}$ controls the entropy of the policy [383, 384]. Q-learning is an iterative approach to obtaining estimates $\hat{Q}$, in which these values are initialized arbitrarily and then updated according to the update rule for given a dataset D of observed state-action-reward-subsequent state tuples (s, a, r, s')

$$\hat{Q}(s,a) \leftarrow \hat{Q}(s,a) + \alpha \left[r + \gamma \text{max}_{a' \in A} \hat{Q}(s',a') - \hat{Q}(s,a) \right]. \tag{6.5}$$

6

6.2.2 Off-policy evaluation

Assessing the performance of some sequential decision-making policy on an observational dataset is known as OPE. OPE can be challenging in practice because the data used in the evaluation is generated with a *behavior* policy that is different from the policy subject to evaluation. Formally, OPE boils down to estimating the value for an evaluation policy V_{π_e} given a dataset of n trajectories $D = \left\{ tr^{(i)} \right\}_{i=1}^{n}$ generated by a distinct behavior policy π_b and trajectories of states, actions and rewards $tr = (s_0, a_0, r_0, \ldots, s_{T-1}, a_{T-1}, r_{T-1}, s_T)$. Various classes of approaches exist to tackle the OPE problem. We here discuss three classes of OPE that are the de-facto standard for RL in healthcare [376]. In general, only evaluation policies with some *support* in the behavior policy can be evaluated, i.e. $\pi_b(a|s) = 0 \implies \pi_e(a|s) = 0$ has to hold.

Per-Horizon Weighted Importance Sampling

Importance Sampling (IS) is a popular class of OPE approaches in which observed returns in historical data are weighted to adjust for differences in π_b and π_e with so-called importance weights for each trajectory $tr \in D$: $\rho_t^{tr} = \prod_{i=1}^{t} \frac{\pi_e(a_i^{tr}|s_i^{tr})}{\pi_b(a_i^{tr}|s_i^{tr})}$. In an episodic setting with varying trajectory lengths, these weights can be normalized to account for varying trajectory lengths to obtain the so-called per-horizon weighted importance sampling (PHWIS)[84]. The state-value estimator for PHWIS is defined as follows for the set of all trajectory lengths $\mathcal{L}$ and their relative occurrence $W_{l \in \mathcal{L}} = \frac{|\{tr_i | T_i = l\} \in D|}{n}$ in the dataset D used for evaluation:

$$\hat{V}_{\pi_e}^{\text{PHWIS}}(D) = \sum_{l \in \mathcal{L}} W_l \sum_{\{tr_i | T_i = l\}} \sum_{t=0}^{T_i - 1} \frac{\rho_t^{(i)}}{\sum_{\{tr_i | T_i = l\}} \rho_t^{(i)}} \gamma^t \, r_t^{(i)}. \tag{6.6}$$

IS methods in general, and PHWIS specifically, are consistent estimators and have low bias and high variance in a comparison to other estimators.

A Diagnostic for Importance Sampling

If the evaluation policy is in low agreement with the behavior policy, the importance weights for a large number of trajectories will be close to 0 and the importance weights for a small number of trajectories will be very large. As a result, the return of a small number of trajectories dominates the estimate $\hat{V}_{\pi_e}$. Confidence in estimates obtained with importance sampling should therefore not only include the total number of samples, but also their weights [130]. A diagnostic known as the effective sample size (ESS) can be used for this purpose [183]:

$$ESS = \frac{1.0}{\sum_i^n \left(\rho_{T_i}^i / \sum_j^n \rho_{T_j}^j\right)^2} \tag{6.7}$$

where $\rho_{T_i}^i, \rho_{T_j}^j$ importance weights for trajectories i, j of length T_i, T_j and n the size of dataset D as defined earlier. An ESS close to n indicates that all trajectories are weighted almost equally while an ESS close to one that most trajectories are weighted close to zero and that a single trajectory has nonzero weight. The latter case can be seen as one in which a single trajectory has an out-sized influence on the estimated performance of π_e. The ESS is (close to) zero if all importance weights ρ_t^{tr} are (close to) zero, i.e. if all trajectories in D have (near-)zero probability of being generated by π_e.

Fitted Q Evaluation

OPE approaches in the second class focus on the usage of regression techniques to more directly estimate value functions V_{π_e} and Q_{π_e} and are therefore known as direct methods (DMs). These estimates can be obtained by plugging estimates of the transition and the reward functions into their definitions or they can be estimated directly. Fitted Q evaluation (FQE) is a biased OPE estimator based on the well-known fitted Q-iteration algorithm [92, 190]. It produces a direct estimate $\hat{Q}_{\pi_e}^{\text{FQE}}$ by casting the OPE problem as an iterative supervised learning problem. FQE has gained popularity due its simplicity, low variance and good empirical performance in small-data regimes [138, 376]. When $\hat{Q}_{\pi_e}^{\text{FQE}}$ has been learned, its state-value equivalent $\hat{V}_{\pi_e}^{\text{FQE}}$ can be derived as:

$$\hat{V}_{\pi_e}^{\text{FQE}}(s) = \sum_{a \in A} \hat{Q}_{\pi_e}^{\text{FQE}}(s, a)\ \pi_e(a|s)\,. \tag{6.8}$$

This state-value estimator can then be applied to all initial states in some evaluation set D to evaluate the overall performance of π_e:

$$\hat{V}_{\pi_e}^{\text{FQE}}(D) = \frac{1}{n} \sum_{i=1}^{n} \hat{V}_{\pi_e}^{\text{FQE}}(s_0^{(i)}) \tag{6.9}$$

Per-Horizon Weighted Doubly Robust Estimator

The third class of OPE approaches combines the first two classes and are therefore known as hybrid methods. We here focus on so-called doubly robust estim-

ators[165]. Doubly robust estimators use estimators $\hat{V}^{\mathrm{DM}}_{\pi_e}$ and $\hat{Q}^{\mathrm{DM}}_{\pi_e}$ obtained with some direct method as covariates in an importance sampling method in order to reduce its variance. In contrast to some other hybrid methods, doubly robust estimators do not require additional tuning of hyperparameters [354]. The quality of doubly robust estimators depends on the quality of the direct method. FQE is a direct method with good results in an extensive empirical evaluation [376]. In this work, we therefore use doubly robust version of the PHWIS estimator (6.6) known as per-horizon weighted doubly robust (PHWDR) estimator and use covariates obtained with FQE. The weighted doubly robust (WDR) estimator as defined by Thomas and Brunskill [354] is:

$$\begin{aligned}\hat{V}^{\mathrm{WDR}}_{\pi_e}(D) =& \frac{1}{n}\sum_{i=1}^{n}\hat{V}^{\mathrm{FQE}}_{\pi_e}(s_0^{(i)}) \\ &+ \sum_{i=1}^{n}\sum_{t=0}^{T}\gamma^t\omega_t^{(i)}\left[r_t^{(i)} - \hat{Q}^{\mathrm{FQE}}_{\pi_e}\left(s_t^{(i)}, a_t^{(i)}\right) + \gamma\hat{V}^{\mathrm{FQE}}_{\pi_e}\left(s_{t+1}^{(i)}\right)\right]\end{aligned} \tag{6.10}$$

where

$$\omega_t^{(i)} = \frac{\rho_t^{(i)}}{\sum \rho_t^{(i)}} \tag{6.11}$$

and can be applied per-horizon following [285]:

$$\hat{V}^{\mathrm{PHWDR}}_{\pi_e} = \sum_{l\in\mathcal{L}} W_l \hat{V}^{\mathrm{FQE}}_{\pi_e}\left(\{tr_i | T_i = l\}\right). \tag{6.12}$$

Note that if the ESS is small, the obtained estimate largely depends on $\hat{V}^{\mathrm{FQE}}_{\pi_e}$ in the first term in (6.10).

6.3 Related Work

6.3.1 Reinforcement learning in the ICU

The medical domain has recently become an important application area of interest for RL due to RLs ability to address sequential decision-making problems that involve a degree of uncertainty [P4]. Within the medical domain, applications in the ICU are particularly of interest because of three key reasons. Firstly, a large number of measurements is collected for patients in the ICU with relatively high frequency. Moreover, various ICU datasets have been made widely available [168, 277, 357]. These datasets can be used to model the patient's condition as an environment state. Secondly, patients in the ICU suffer from conditions that require a relatively high degree of physiological control. Taken jointly with the large number of measurements, this high degree of control suggests that treatment outcomes are mostly dependent on factors known to and in control of caregivers. From a learning perspective, this limits the

amount of unexplainable variance in patients outcomes based on treatments, i.e. the stochasticity of the environment is limited and the learning problem may be reasonably tractable in comparison to some other medical settings. Finally, the reward signal in the ICU can be defined in terms that lie within a reasonable time frame and can be easily quantified such as survival at ICU discharge and at 90 days after admission.

Komorowski et al. [181] introduced an approach to the treatment of sepsis based on tabular Q-learning with a discrete state space obtained by $k-$means++ clustering and with a discretized action space of vasopressor and IV fluid dosage obtained by binning. The study on sepsis also produced recommendations on using RL in an ICU setting which were adopted in the present work [130] . Subsequently, Roggeveen et al. [298] studied the transferability of sepsis treatment policies across patient populations and proposed a deep RL based approach.

An approach to quantify and improve guideline compliance of RL-based sepsis treatment based on reward shaping was proposed by Festor et al. [104]. The approach relies on case-specific scenarios defined by experts and does not guarantee adherence to constraints, whereas our approach leverages existing knowledge encoded in clinical guidelines and guarantees compliance to constraints on the action space. Another domain-specific safety-aware approach was proposed by Jia et al. [164]. It focuses on the temporal nature of constraints, a feature that our approach supports by using e.g. an LTL encoding of constraints [P1]. Other technical extensions in the sepsis treatment domain include the usage of a continuous state space [286], clinician-in-the-loop decision-making with set-valued policies [347] and diverse policies [110], and the incorporation of partial observability of the true patient state [109]. Our approach complements these works by ensuring that only guideline-compliant actions are selected by the resulting policy.

We evaluate our approach on the problem of optimizing MV settings as introduced by Peine et al. [266]. We extend their Q-learning-based approach with knowledge from guidelines and contribute additional OPE evaluations to theirs following recommendations by Gottesman et al. [130]. Chen et al. [57] studied the use of a hybrid RL approach to find optimal MV settings. Results show state-of-the-art performance in a simulated environment, an evaluation on real-world data and the incorporation of safety constraints are left for future work. Prasad et al. [281] and Yu, Liu and Zhao [401] studied the problem of MV "weaning", i.e. of decreasing the degree of ventilator support and training the patient to eventually be extubated. One of these proposed to use an inverse RL approach [401]. Inverse RL is related to the imitation learning approach included in our evaluations.

6.3.2 Reinforcement Learning with Instructions and Constraints

The literature on RL with instructions and (safety) constraints is vast. Therefore, we focus on closely related works, i.e. works that use symbolic instructions

and constraints. Guidelines are best represented symbolically since symbolic formalisms are composable, have unambiguous semantics and allow for reasoning [292]. Since these are most closely related to constraints from guidelines. For a more comprehensive overview see [114, 211, 248].

A recent string of works has looked into the usage of symbolic safety constraints for RL based on the concept of automata from the discipline of formal methods [5, 75, P1, 398]. These works target safety constraints with a temporal component whereas we target more basic constraints that do not depend on time. These approaches, however, can be used when guideline instructions include a dependence on time e.g. 'never administer drug X if patients previously showed an allergic response of type Y'.

Another string of works uses symbolic instructions to divide the full task into smaller subtasks and alter the reward function in order to increase data efficiency [9, P3, 158, 160]. Our framework similarly alters the reward function using reward shaping but includes the operationalisation of knowledge in medical guidelines rather than a generic symbolic subtask decomposition.

6.4 Guideline-informed Reinforcement Learning

The proposed framework for guideline-informed RL generates policies based on both knowledge encoded in guidelines and from experiences obtained in clinical practice. It differs from the standard RL setup from Section 6.2.1 in two ways as visualized in Figure 6.1. Both extensions to the general RL framework are informed by clinical guidelines, but the way in which these guidelines are used to inform the learner differs between these two extensions.

The first extension consists of an action filter that forces the agent to only select treatment actions in accordance with the guideline. This extension allows the usage of explicit treatment advice based on parameters describing the patient state. Explicit advice on treatment decisions is the dominant type of knowledge in the computer-interpretable guidelines and can be found in the form of ontologies, decision tables and logic [292, 350]. We propose to encode treatment advice as a filter that removes treatment actions that are not in line with the guideline from the agent action space as described in Section 6.4.2.

The second extension consists of an approach to informing the agent of desirability of properties in the patient condition with reward shaping. We include this second extension to model guideline aspects that cannot be enforced as hard constraint on the action space. Guidelines often contain particular target values for specific indicators. These target values are designed to help clinicians in assessing the patient state at a particular point in time and they are often derived from evidence that links these target values to better patient outcomes. These target values are therefore a natural fit for reward shaping. These target values cannot be modeled as hard constraints. Firstly, because patients are typically in a condition that would violate a hard constraint at treatment onset. Secondly, because many indicators can only be meaningfully

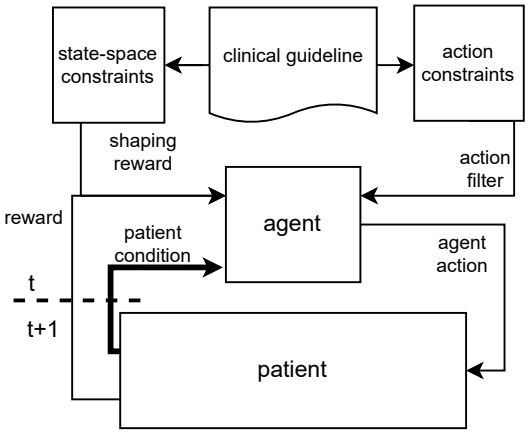


Figure 6.1: Overview of the guideline-informed RL approach. Clinical guidelines are first encoded into state-space constraints and action constraints. Action constraints describe allowable treatment decisions and are strictly enforced with a filter that removes all non-compliant treatment decisions from the agent's action space. State-space constraints describe desirable properties in the patient condition. The learning agent is informed of state-space constraints with additional, shaping rewards.

controlled indirectly or in a delayed fashion due to nature of physiological processes. Therefore, we alter the reward function to include generic knowledge on treatment quality and the human physiology in the learning process [81]. We detail both extensions in this section but first introduce how guidelines are formalized in the proposed framework.

6.4.1 Formalisation of guidelines

We consider a set of l variables $\mathcal{V} : \{\nu_1, \ldots, \nu_l\}$ to describe patient states and treatment decisions and a finite set of m ranges to describe allowable ranges $\mathcal{R} = \left\{\nu_1 \in [v_{\min}^{(1)}, v_{\max}^{(1)}], \ldots, \nu_1 \in [v_{\min}^{(i)}, v_{\max}^{(i)}], \ldots, \nu_j \in [v_{\min}^{(m)}, v_{\max}^{(m)}]\right\}$ for these variables. We consider each clause φ as a subset of the power set of ranges $\varphi \subseteq 2^{\mathcal{R}}$. We require that all values fall within the provided bounds in φ to comply to that clause. Formally, a set of measurements $\{\nu_1 = v_1, \ldots, \nu_n = v_n\} \models \varphi \iff \bigwedge_{j=0}^{|\varphi|} \left(v_i \in \left[v_{\min}^{(j)}, v_{\max}^{(j)}\right] \vee \nu_i \neq \nu_j\right)$ where $|\varphi|$ denotes the number of ranges in φ. Note that multiple ranges can be assigned to a single variable ν within the guideline $\mathcal{R}$ but that these ranges should overlap.

We continue by connecting this formalisation of the guideline to the RL framework. Specifically, we assume access to a set of measurable features of the true patient state s_{true} via a set of $k < l$ basis functions $\{\phi_1, \ldots, \phi_k\}$ such that we obtain a feature vector representation $\vec{\phi}_S(s_{\text{true}}) = (\phi_1(s_{\text{true}}), \ldots, \phi_k(s_{\text{true}}))$ for all $s_{\text{true}} \in S$. Since we can only access the vector representation, we let s

refer to vector representations of true states in the remainder of this work. Additionally, we assume a similar vector representation $\vec{\phi}_A = (\phi_{k+1}(a), \ldots, \phi_l(a))$ for all $a \in A$ of the action space. We finally assume that these basis functions correspond to the variables $\mathcal{V}$ such that the concatenated representation $\vec{\phi}(s_{\text{true}}, a) = \vec{\phi}_S(s_{\text{true}}) \oplus \vec{\phi}_A(a)$ of a state-action pair produces a set of measurements $\{v_1, \ldots, v_l\}$ for variables $\{\nu_1, \ldots, \nu_l\}$. We continue by elaborating how the formalized guideline can be incorporated into RL approaches.

6.4.2 Guideline-based action filter

Explicit treatment advice is encoded as a hard constraint on the agent action space. This constraint specifies which actions are allowable and is enforced by removing all actions that are not allowable from the action space. In this section, we formally introduce the action filter and its components.

The action filter is formalized as a function $\mathcal{C}_A : A \times S \to \{0, 1\}$ that determines which actions comply to a guideline for a given state. Guidelines describe a set of generally allowable treatment decisions for a given patient state and leave the final particular decision up to the clinician. The action space constraints in the guideline are therefore modeled using a disjunction of conjunctions (disjunctive normal form or DNF) in our framework, see Equation 6.18 below. That is, a state-action pair is compliant if any of the clauses allows for it. Note that the clauses themselves may contain a conjunction so that the framework allows for bounds within which any decision is allowable, bounds that have to be met all times and bounds that are contingent on other bounds. For a guideline consisting of n clauses φ_i each possibly consisting of multiple bounds, we require:

$$\mathcal{C}_A(a|s) = \bigvee_{i=1}^{n} \left[\vec{\phi}(s, a) \models \varphi_i \right]. \tag{6.13}$$

We denote the set of guideline-compliant actions for a particular s as $A_{\mathcal{C}}(s) \subseteq A : \{a \in A | \mathcal{C}_A(a|s) = 1\}$.

We now consider two approaches to enforcing the compliance constraints in RL policies. The first approach consists of enforcing the constraints after learning. Specifically, we derive a constrained policy $\pi_{\mathcal{C}}$ from an arbitrary policy π by only allowing guideline-compliant actions:

$$\pi_{\mathcal{C}}(a|s) = \begin{cases} \dfrac{\pi(a|s)}{\sum_{a' \in A_{\mathcal{C}}(s)} \pi(a'|s)} & \text{if } a \in A_{\mathcal{C}}(s) \\ 0 & \text{otherwise.} \end{cases} \tag{6.14}$$

The approach based on Equation 6.14 removes noncompliant actions from an existing policy. The resulting policy is guaranteed to adhere to the action constraints $\mathcal{C}_A$ as a result. Furthermore, the approach is applicable to imitation learning and does not require any additional training, which can be computationally costly in practice.

However, there are several downsides: firstly, if the policy π never selects a compliant action for some state s, the denominator in Equation 6.14 may be zero for that state. In this case, some fallback policy has to be used. Secondly, non-compliant actions may have been used in the construction of π. For example in Q-learning, if non-compliant actions would be included in the calculation of the Q-function estimates $\hat{Q}$, applying Equation 6.14 to this policy would result in estimated Q-values that are not reflective of the compliant policy $\pi_{A_\mathcal{C}}$ which may be suboptimal as a result. We therefore consider a second approach that includes the action filter in the learning process.

The second approach incorporates the action filter in the estimation of Q-values. This requires the constraints to be available at training time but ensures that the resulting estimates $\hat{Q}$ are according to the constrained policy $\pi_\mathcal{C}$ rather than according to an unconstrained policy. The action filter is incorporated in the Q-function update rule in Equation 6.5:

$$\hat{Q}(s,a) \leftarrow \begin{cases} \hat{Q}(s,a) + \alpha \left[r + \gamma \max_{a \in A_\mathcal{C}(s')} \hat{Q}(s',a) - \hat{Q}(s,a) \right] & \text{if } a \in A_\mathcal{C}(s) \\ -\infty & \text{otherwise.} \end{cases} \tag{6.15}$$

Policies can be derived traditionally with Equations 6.3 and 6.4 and are guaranteed to comply to the guideline if the training data contains at least one compliant action for every state.

6.4.3 Guideline-based reward shaping

Implicit clauses in a guideline describe preferable properties in the patient's condition. Because a patient's condition can be very poor at the onset of treatment and deteriorate stochastically, these constraints cannot be enforced as hard constraints. Instead, the learner is informed of preferable conditions with an altered reward function. Specifically, we inject additional rewards based on a compliance metric $\mathcal{C}_S : S \to \mathbb{R}$ defined over the RL state space. While various aggregations can be used, we here define state compliance as the average clause satisfaction rate

$$\mathcal{C}_S(s) : \frac{\sum_i^n s \models \varphi_i}{n} \tag{6.16}$$

for all clauses defined over the state space.

The compliance metric $\mathcal{C}_S$ was then used to inform the learner following the convention of potential-based reward shaping [254]. In potential-based reward shaping, a transformation is applied to the reward function in such a way that any optimal policy under the transformed reward R^+ is also optimal under the original reward function [135, 254]. A shaped reward function $R^+(s,a,s') : R(s,a) + \gamma\Phi(s') - \Phi(s)$ is used with so-called potential functions Φ:

$$\Phi(s) : \begin{cases} 0 & \text{if } s \text{ a terminal state} \\ \mathcal{C}_S(s)c & \text{otherwise} \end{cases} \tag{6.17}$$

where $c \in \mathbb{R}$ a scalar to balance environment and shaping rewards.

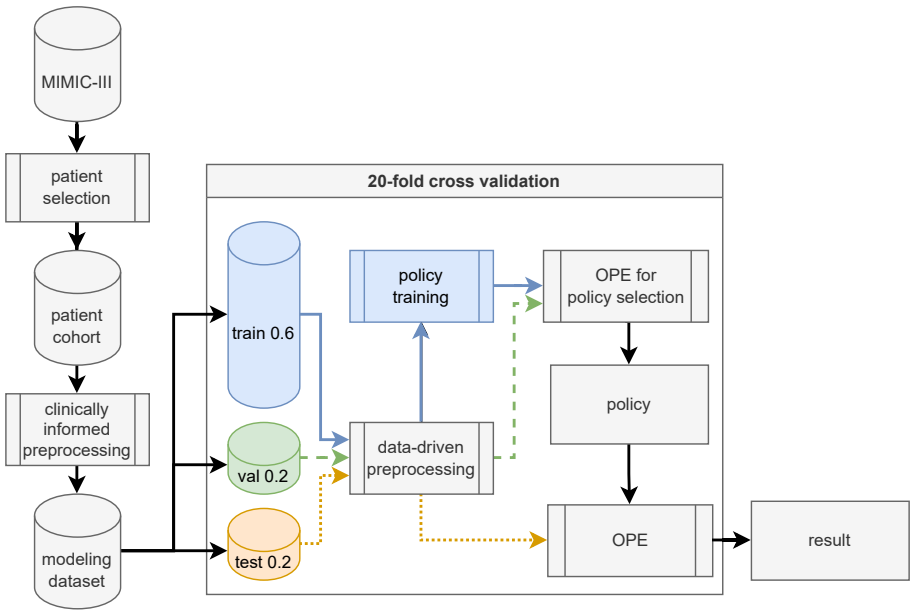


Figure 6.2: Outline of the study design.

6.5 Materials and methods

We employ a retrospective study design summarized in Figure 6.2. The design consists of steps for patient cohort and data selection, data pre-processing, guideline-informed policy learning and off-policy evaluation. All steps were performed on the MIMIC-III v1.4 database [168] and best practices for reproducibility in machine learning research were followed [274]. The data selection, pre-processing, policy learning and off-policy components are based on prior work by Peine et al. [266] and code was reused where available. We detail each step in the study design below, reiterating the steps copied from [266] for completeness.

6.5.1 Patient cohort and data selection

Patient cohort and data selection was based on Peine et al. [266]. Out of all 61,532 admissions in the MIMIC-III v1.4 database, a total of 10,597 MV events (see below for the definition of an MV event) of 9,355 unique patients were extracted. Admissions were selected using the following inclusion criteria: patient age of 18 or higher at the moment of admission, treatment not withdrawn in assessed time-frame, documented 90-day mortality, mechanical ventilation of at least 24h and a documented set tidal volume. Cohort characteristics are summarized in Appendix C:.

For each admission, data was aggregated into 4-hour time windows for 4h

prior to and 72h after the onset of ventilation. Variables were aggregated as appropriate following [266]. The onset of ventilation was derived from a documented value for the variable 'set tidal volume' Vt_{set}. The presence of either a value for variables Vt_{set}, PEEP or FiO_2 during an 8h time window continued the ventilation event, while it was discontinued by the documentation of extubation, initiation of non-invasive ventilation and/or supplemental oxygen supply. For each admission, only the first ventilation event was included resulting in a total of 10,581 included ventilation events and 165,275 ventilation decisions.

6.5.2 Pre-processing

The pre-processing step consisted of imputation of the missing data, scaling and clustering phases similar to Peine et al. [266]. The data imputation phase consisted of two steps: in the first step, missing data was imputed with a sample-and-hold scheme with clinically informed time windows, see Tables C.2 and C.3 in Appendix C for details. After this phase, admissions with more than 50% missing values were removed from the dataset, resulting in a removal of less than 1% of the data. Next, data was separated into train, validation and test sets to ensure the validity of subsequent phases. Twenty-fold random permutation cross validation (RPCV) was employed to enable estimation of spread of the OPE result while maintaining sufficiently sized data splits, see Section 6.5.4 [274]. The imputation phase continued with a second imputation step. In this step, all variables were first centered around the mean, scaled to unit variance and then imputed. Imputation was performed using k-nearest neighbor imputation with $k = 5$ and a Euclidean distance metric robust to the presence of missing values. The transformations performed on the training set in this second imputation phase were stored and then applied to the test and validation sets to protect their unseen nature and the validity of results.

Patient demographics and clinical variables that describe the patients' condition were clustered into discrete states to be used by the RL algorithm, again inspired by [266]. Specifically, k-means clustering was performed with $k = 650$. The documented treatment decisions were also transformed into discrete actions. The treatment decisions in scope for this research were documented by three variables: Vt_{set}, PEEP, and FiO_2. These action variables together make up a particular configuration of a mechanical ventilator at each point in time. We transformed this three-dimensional action space into a one-dimensional discrete action space. First, all values were grouped into 7 bins for all three action variables as described in Table C.4 in Appendix C. All combinations of binned settings were then assigned unique action identifiers which resulted in a final discrete action space of cardinality $7^3 = 343$.

6.5.3 Guideline encoding

The guideline implemented to illustrate our approach is the strategy for protective mechanical ventilation originally developed for patients suffering from

Space	Variable	Guideline	Constraint	#
State	Pplat	≤ 30	≤ 30	φ_1
	pH	$7.3 - 7.45$	$\in (7.2, 7.5)$	φ_2
	RR	$6 - 35$	≤ 35	φ_3
	SpO_2	$88 - 95\%$	≥ 88	φ_4
Action	Vt_{set}	6 (initial)	≤ 8.5	φ_5
	FiO_2	0.3 and 5	$FiO_2 \in [0.3, 0.5) \wedge$ PEEP$= 5$	φ_6
	and	0.4 and 5		
	PEEP	0.4 and 8	$FiO_2 \in [0.4, 0.6) \wedge$ PEEP$\in [4, 8]$	φ_7
		0.5 and 8		
		0.5 and 10	$FiO_2 \in [0.5, 0.7) \wedge$ PEEP$\in [8, 10]$	φ_8
		0.6 and 10		
		0.7 and 10	$FiO_2 \in [0.7, 0.8) \wedge$ PEEP$\in [10, 14]$	φ_9
		0.7 and 12		
		0.7 and 14		
		0.8 and 14	$FiO_2 \in [0.8, 0.9) \wedge$ PEEP$= 14$	
		0.9 and 14	$FiO_2 \in [0.9, 1.0) \wedge$ PEEP$\in [14, 18]$	φ_{10}
		0.9 and 16		
		0.9 and 18		
		1.0 and 18	$FiO_2 = 1.0 \wedge$ PEEP$\in [18, 24]$	φ_{11}
		1.0 and 20		
		1.0 and 22		
		1.0 and 24		

Table 6.1: Target values for state- and action-space constraints. We pair the original guideline values with their formalisation as constraints. Pplat: plateau pressure in cmH_2O, pH: acidity of blood, RR: respiratory rate in breaths/min, SpO_2: O2 saturation pulseoxymetry. Vt_{set}: set tidal volume in ml/kg IBW (ideal body weight, also known as predicted body weight), FiO_2: fraction of inspired oxygen, PEEP: positive end-expiratory pressure in cmH_2O.

acute hypoxemia associated with diffuse opacities on lung imaging due to varying etiologies of non-cardiac origin luding diffuse inflammation known as the Acute Respiratory Distress Syndrome (ARDS). This strategy is based on the low tidal volume group from the ARDSnet ARMA trial which aimed to achieve sufficient arterial oxygenation and avoid respiratory acidosis while protecting the lung from traumatic distention by avoiding high tidal volumes [251]. The low tidal volume strategy resulted in marked reductions in mortality before hospital discharge and a marked increase in ventilator free days until day 28 as compared to a higher tidal volume strategy. In both groups, the same sliding scale table was used to set positive end expiratory pressure (PEEP) and the fraction of inspired oxygen concentration (FiO2). As such, this sliding scale of allowed combinations of PEEP and FiO2 were part of this protective ventilation strategy for ARDS patients that has since extended to all mechanically ventilated patients [98, 128, 308]. Therefore, this strategy is a logical starting point for our use case, although it should be noted that optimal ventilator settings continue to be subject of intense debate and various other approaches exist (PMID: 17417980 and PMID: 28828363 and PMID: 29043834).

The guideline was encoded into the constraints listed in Table 6.1 in collaboration with medical experts. Specifically, the strategy informs on treatment decisions both explicitly and implicitly. Explicit clauses $\varphi_5 - \varphi_{11}$ advise specific allowable treatment decisions. In this case, these are given as upper and lower bounds for mechanical ventilator settings and were encoded as constraints on the RL action space as detailed in Section 6.4.2. Implicit clauses $\varphi_1 - \varphi_4$ describe desirable properties in the patients' condition: in this case, these are target values for various measurements obtained at the bedside. These implicit clauses were used for reward shaping as detailed in Section 6.4.3.

The guideline should be interpreted as a set of recommendations rather than a strict directive. Therefore, some of the bounds from the original strategy were relaxed in the constraint encoding in consultation with clinicians. Finally, we highlight that clinical adoption of medical guidelines is generally slow and therefore implementation of the recommendations from our example strategy conincided with the time window in which the dataset was collected. As a result, the dataset can be expected to contain both patients for which the recommendations were followed and patients for which other choices were made.

In this case study, compliance of actions is independent of the state. As a result, the compliance function $\mathcal{C}_A$ could be evaluated independently from the state for each action in the action space. By doing so for all 343 actions in our use case, 72 allowable actions were obtained. For the constraints listed in the bottom of Table 6.1, the resulting action space constraint in DNF is:

$$\begin{aligned}\big(&(\mathrm{Vt}_{set} \leq 8.5 \ \wedge \mathrm{FiO}_2 \in [0.3, 0.5) \wedge \mathrm{PEEP} \in [5, 5]) \vee \\ &(\mathrm{Vt}_{set} \leq 8.5 \ \wedge \mathrm{FiO}_2 \in [0.4, 0.6) \wedge \mathrm{PEEP} \in [4, 8]) \vee \\ &\dots\big)\end{aligned} \tag{6.18}$$

for clauses $\varphi_5 - \varphi_1 1$ in Table 6.1.

Abbr.	Description	Policy	Constraints
O	Returns/actions in test set	–	–
IL	Imitation Learning, mimick clinicians policy	Eq. 6.19	Policy
QL_S	Q-learning, stochastic	Eq. 6.4	Policy, Q-function
QL_D	Q-learning, deterministic	Eq. 6.3	Policy, Q-function

Table 6.2: Algorithms included in the evaluation. Constraint variants 'Policy' and 'Q-function' refer to Equations 6.14 and 6.15 respectively.

6.5.4 Evaluation

We compare all algorithms in Table 6.2 and vary the way in which constraints are enforced, i.e. directly in the policy with (6.14) or in the Q-function with (6.15) where applicable. We include an unconstrained, vanilla baseline for all approaches. IL policies were inferred directly from the train set D:

$$\hat{\pi}(a|s) = \frac{|(s,a)|}{|s|} \sim D. \tag{6.19}$$

We use an unconstrained IL policy as a fallback when no compliant action is available for all policies.

For the evaluation of the resulting policies, we use OPE on a held-out test set. The held-out test set was preprocessed similarly to the training and validation data. Specifically, data transformations for imputation and clustering that were obtained by fitting to the training set were applied to this held-out set to ensure validity of the results [274]. The expected return obtained with OPE is directly related to mortality and higher expected return relates to lower expected mortality.

To assess generalizability of the result, we repeat the experiment for twenty separate splits of train, validation and test data with RPCV, also known as 'shuffle and split'. Samples are first shuffled and then split into a tuple of train, validation and test splits for a given number of iterations. This allows for a large number of repeated experiments (here: 20) while retaining reasonable proportions of samples on all sides of the splits (here: train=0.6, validation=0.2, test=0.2).

For the off-policy evaluation, the methods FQE, PHWIS and PHWDR introduced in Section 6.2 were used. For PHWDR, we used FQE for estimates $\hat{V}(s_0)$ and $\hat{Q}(s,a)$. We report the effective sample size (ESS) as a diagnostic for the quality of the PHWIS and PHWDR estimators. We evaluate compliance of the resulting policy by reporting the probability that the policy takes an action that is allowed by the guideline. Specifically, for an evaluated policy π_e we report the probability of taking a compliant action:

$$P(a \in A_{\mathcal{C}}) = \frac{\sum_{s \in S} \sum_{a \in A_{\mathcal{C}}(s)} \pi_e(a|s)}{|S|}. \tag{6.20}$$

For all metrics, we report mean and 95% bootstrap confidence intervals of the mean obtained with 20-fold RPCV.

Implementation details

All experiments were run on a machine with an AMD Ryzen 7 4800U CPU and Ubuntu 22.04.2 (64bit), CPython v. 3.11.0, Scikit-learn v. 1.2.0 [265], Pandas v1.5.2, Numpy v.1.23.1 [140], Scipy v.1.10.0 [375], Postgres v14.7. Tabular Q-learning was used for estimation of all Q-functions and FQE was run for 50 iterations.

6.6 Results & Discussion

Figure 6.3 shows a comparison between all included approaches. We first look at the compliance results (top right) and see that both the 'Policy' and 'Q-function' variants produce more compliant decision-making in comparison with the clinicians behavior ('O' and 'IL'+'Unconstrained'). Out of the constrained approaches, the 'Q-function' approach is preferable from the perspective of compliant decision-making as it produces fully compliant policies. The 'Policy' variants are not fully compliant. This is explained by a total probability mass of zero for the compliant actions for some states, in which case an unconstrained IL policy is used as a fallback.

We now turn to the expected return as obtained with the three OPE approaches in the left-hand column of Figure 6.3 and start with the model-based FQE evaluation (top right). We see that QL_D outperforms the clinicians decision-making here: expected returns for QL_D are significantly higher than those observed (O) in the test set and those estimated for a policy that mimicks clinicians (IL) across compliance variants. Out of all QL_D variants, the Q-function variant yields the best policy as it is both more compliant and significantly outperforms the Policy variant (Wilcoxon signed-rank test, $p < 0.001$).

Moving on to results obtained with the PHWIS estimator (center left), we see results with high variance. We first investigate the results for QL_D, where we see missing results for the Unconstrained and Q-function variants and highly varied results for the Policy variant. The explanation for these results can be found in the figure showing effective sample sizes (center right): effective sample sizes are (near-) zero for all QL_D variants, indicating a downside of IS-based evaluations when evaluation and behavior policies differ too much. Results for QL_S show that stochastic policies suffer less from this issue. Stochastic policies assign nonzero probability to multiple actions and are therefore more likely to agree with the behavior policy, more likely to produce higher effective sample sizes which may result in lower variance for IS-based evaluations.

Continuing our analysis of the expected returns obtained with PHWIS, we see that all compliant variants of the learning-based approaches IL and QL_S perform comparable to the results observed in the test set, see Table C.5 in Appendix C for significance results. More so, we see that compliance constraints

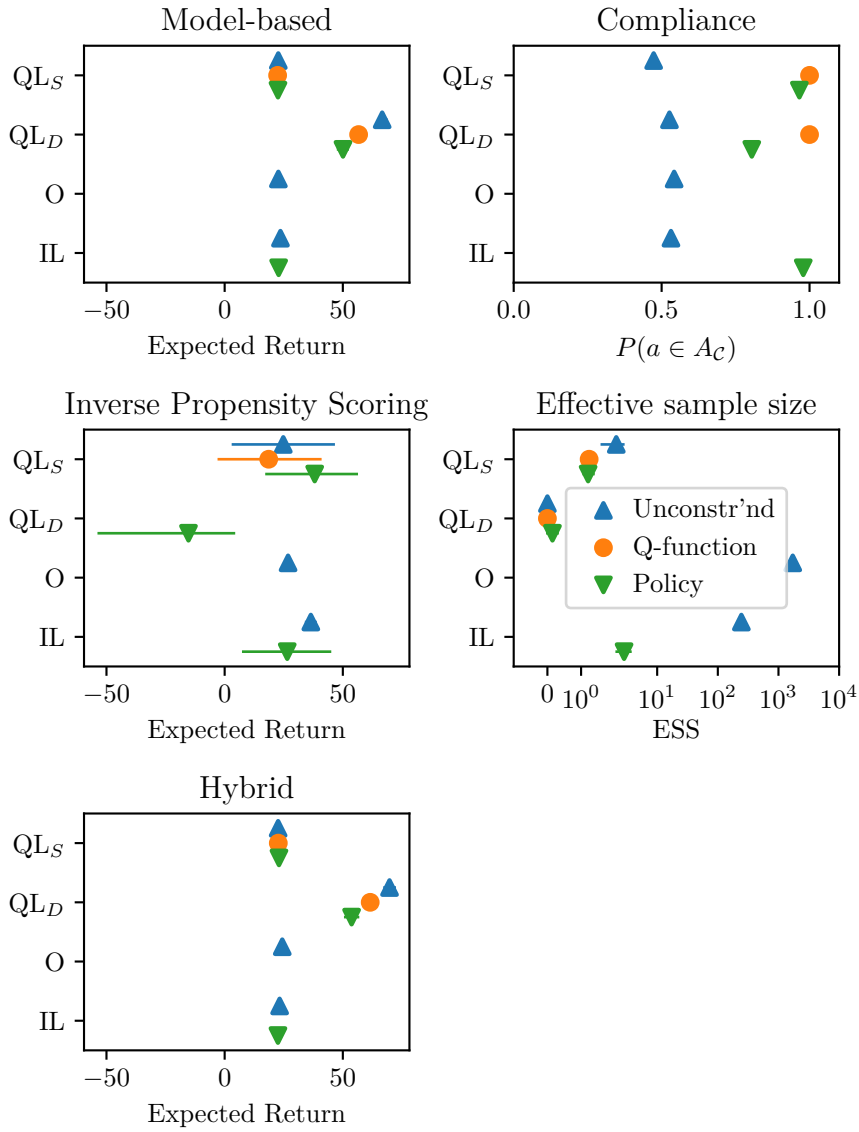


Figure 6.3: Expected return obtained with various OPE approaches (left-hand column), the probability of an noncompliant action (top right) and the effective sample size (bottom right. All figures show the 95% CI of the mean across 20 folds.

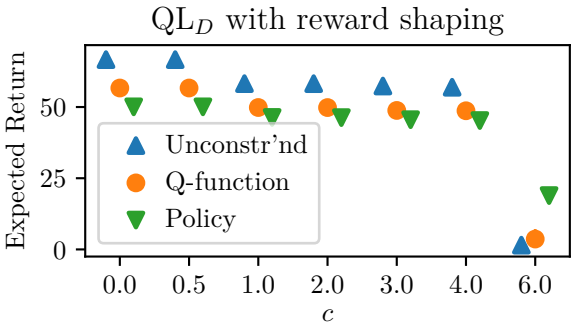


Figure 6.4: Mean expected return for various values of shaping reward scalar c obtained with deterministic Q-learning and FQE on the test set.

do not negatively affect performance. These promising results are further supported by results obtained with PHWDR (bottom left) and with less variance than PHWIS as a result of the hybrid nature of the PHWDR estimator. We note that the PHWDR results for QL$_D$ fully depend on FQE due to an effective sample size of zero as detailed in Section 6.2.2.

We continue by investigating the effects of reward shaping. Figure 6.4 shows expected returns for various values of the parameter c, which balances shaping rewards and environment rewards. We find that, in general, shaping rewards affect expected returns negatively. Expected returns deteriorate sharply for $c = 6.0$. The maximum obtainable shaping reward under this regime was 120: $c * \text{max time steps} = 6.0 * 20 = 120$. Since this is higher than the maximum obtainable environment reward of 100, the balance between environment and shaping rewards for this choice of c is poor. The average compliance rate across states in the data set is high at 0.978. However, results for $c = 6.0$ (rightmost) show that reward shaping can significantly impact the learning in this data set. The results on shaping rewards in Figure 6.4 were consistent across learning algorithms and constraint variants.

We continue to analyze the decision-making of the obtained policies in Figure 6.5, where the number of selected actions in the test set is visualized. We compare clinicians' decisions observed in the test set to decisions made by unconstrained and constrained 'Q-function' variants of QL$_D$, i.e. the best performing condition according to a model-based evaluation. The Q-learning policies (center and bottom) select actions that are more varied than the clinicians actions (top). Additionally, we see that a compliant policy selects similar parameter settings to the vanilla policy while avoiding particular extreme settings.

We conclude this section with some suggestions from improvement. Firstly, a different encoding of the guidelines can be used. For example, an upper bound for the SpO_2 variable could be provided to discourage high settings for FiO_2 and PEEP. How a guideline should be encoded depends on the clinicians' preferences. We strongly encourage encoding guidelines in consultation

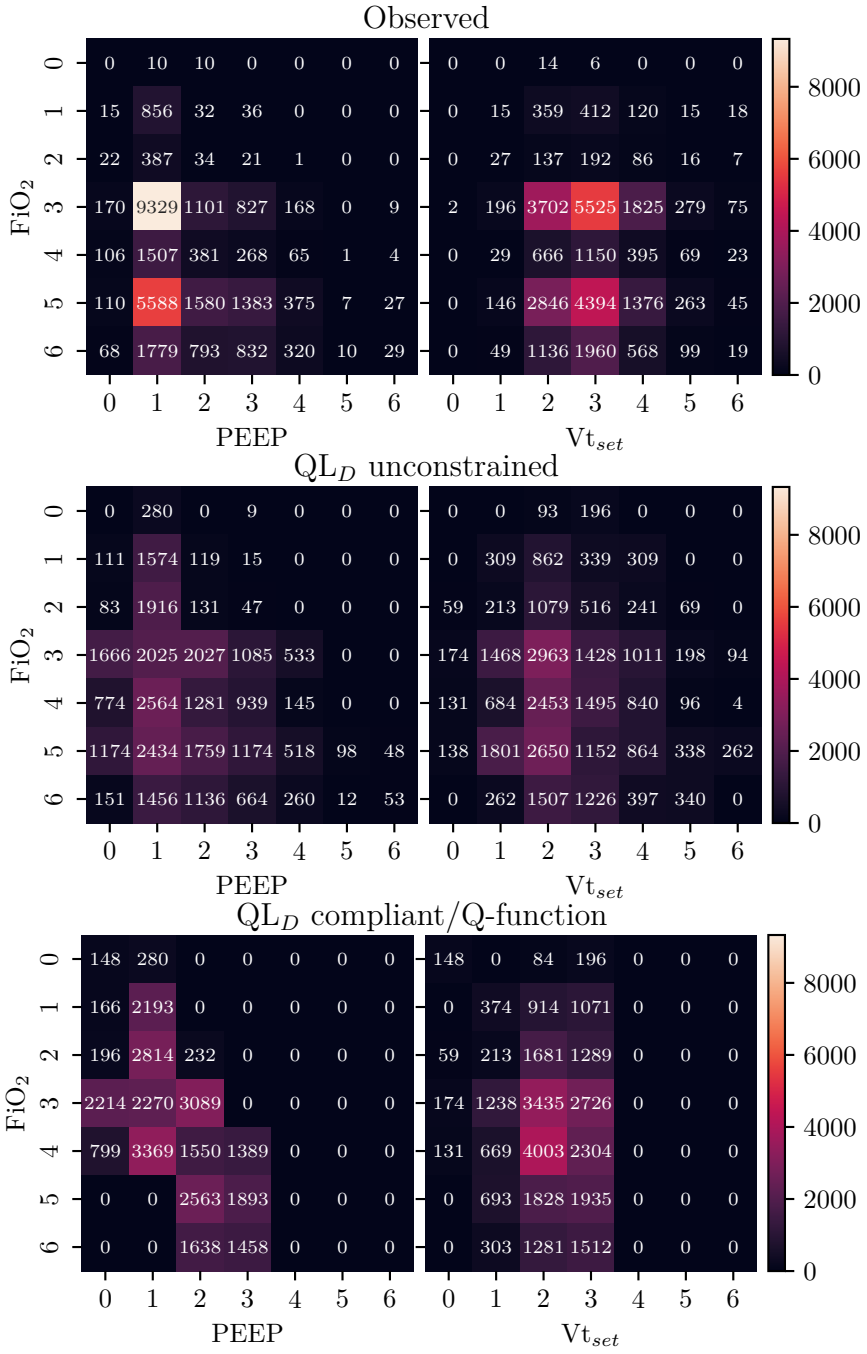


Figure 6.5: Selected actions on the test set represented as three-dimensional binned settings: observed in the test set (top), selected by deterministic and unconstrained Q-learning (center) and constrained Q-learning (bottom).

with clinicians as this involves interpreting the guideline and argue that the presented framework is sufficiently flexible for encoding different interpretations. Secondly, we based our evaluation on an existing RL model to focus our study on the proposed framework rather than on modeling specifics. However, some of the modeling decisions could be improved. We specifically note the inclusion of variables for pulmonary compliance, minute volume and C-reactive protein (crp) when the dataset contains these and the exclusion of variables with similar information content, such as calculated carbon dioxide and $PaCO_2$. Both of these directions for improvement do not reflect on the core contributions of this work directly, but may show improvements in policy behavior and performance.

6.7 Conclusion

We proposed and evaluated a hybrid learning- and knowledge-driven framework for automated clinical sequential decision-making. A knowledge-driven component models a medical guideline and informs the learning-driven component in two ways: by constraining the action space and with an additional reward signal. We implemented our framework by extending an existing model for MV in the ICU and evaluated it using off-policy evaluation. We compared implementations of the proposed framework with varied action constraint enforcement and a varied balance between environment and additional rewards.

We found that our approach produces policies that comply to the medical guideline while outperforming clinicians in terms of expected mortality in a model-based evaluation. In this evaluation, compliant policies are slightly outperformed by non-compliant policies, but compliant policies avoid extreme settings and may hence be more trusted in practice. We found no benefits of including an additional reward signal, indicating that the training data was sufficiently rich per se or that the additional reward does not aid in learning how to achieve the main objective of minimizing 90-day mortality in general.

Our framework can extend existing studies into the use of RL in the medical domain with guideline compliance guarantees and is therefore an important stepping stone in the adoption of RL in clinical practice. It, furthermore, offers opportunities for further research into the representation of guideline constraints, the evaluation of policies when these vary from current policies and hybrid decision-making approaches that combine knowledge with data.

7

Planning for Potential: Efficient Safe Reinforcement Learning

Deep reinforcement learning (DRL) has shown remarkable success in artificial domains and in some real-world applications. However, substantial challenges remain such as learning efficiently under safety constraints. Adherence to safety constraints is a hard requirement in many high-impact application domains such as healthcare and finance. These constraints are preferably represented symbolically to ensure clear semantics at a suitable level of abstraction. Existing approaches to safe DRL assume that being unsafe leads to low rewards. We show that this is a special case of symbolically constrained RL and analyze a generic setting in which total reward and being safe may or may not be correlated. We analyze the impact of symbolic constraints and identify a connection between expected future reward and distance towards a goal in an automaton representation of the constraints. We use this connection in an algorithm for learning complex behaviors safely and efficiently. This algorithm relies on symbolic reasoning over safety constraints to improve the efficiency of a subsymbolic learner with a symbolically obtained measure of progress. We measure sample efficiency on a grid world and a conversational product recommender with real-world constraints. The so-called Planning for Potential algorithm converges quickly and significantly outperforms all baselines. Specifically, we find that symbolic reasoning is necessary for safety during and after learning and can be effectively used to guide a neural learner towards promising areas of the solution space. We conclude that RL can be applied both safely and efficiently when combined with symbolic reasoning.

Based on [P1]:
Floris den Hengst, Vincent François-Lavet, Mark Hoogendoorn, and Frank van Harmelen
Planning for potential: efficient safe reinforcement learning
Machine Learning, 2022

7.1 Introduction

Reinforcement learning (RL) provides an elegant framework for decision making in autonomous agents. In the RL framework, an agent acts in an environment in order to collect rewards [338]. RL driven by neural network-based function approximation, commonly known as Deep Reinforcement Learning (DRL), has recently shown remarkable progress in diverse areas such as personalization (see Chapter 2), robotics [136] and game-playing [148, 321]. The application of DRL in real-world scenarios, however, remains challenging. Despite the significant efforts on making RL agents safe, one of the key challenges remains how to impose "safety constraints that should never [...] be violated" [86].

Safety constraints are present in various high-impact domains such as healthcare and finance. Here, regulations and guidelines describe what behaviors are allowed and disallowed. Typically, the behaviors are not listed explicitly, but described by conditions that have to be met at all times. As such, regulations and guidelines form a symbolic and high-level specification of safe behavior in a particular domain. In many thus governed domains, provable compliance to these specifications is an essential prerequisite for the deployment of any system, including DRL-based ones.

Provably safe DRL has recently been approached from the perspective of symbolic reasoning. Symbolic reasoning provides powerful modeling capabilities, unambiguous semantics and well understood computational properties. [108] introduced a framework for checking a bounded safety constraint on a learned model with probabilistic guarantees. [385] proposed a method for strict adherence, which was extended by [169] to stochastic settings. [5] proposed a mechanism that scales to large state-action spaces using a precomputed *shield* which removes actions iff these are unsafe based on work by [34].

The above works contain proofs of adherence to the specifications but only target a special case in which being safe is correlated with high expected total rewards. A correlation between high expected total rewards and being safe, however, is not present many important application domains. On the contrary, high rewards can be obtained by engaging in disallowed behaviors in many domains. In such domains, regulations are typically put in place precisely to avoid behaviors that yield high reward but come rare but highly undesirable events, negative long-term consequences and negative externalities. For example, guidelines in healthcare protect organs from damage inflicted during treatment in order to safeguard post-treatment quality of life. Although the problem setting of safe RL in the presence of *antagonistic* constraints has been identified before by e.g. [184], it remains, to the best of our knowledge, largely unexplored how these constraints affect performance and how to mitigate negative effects.

In this work, we identify that safe policies do not outperform unsafe policies in terms of expected reward. We theoretically analyze how symbolic safety constraints impact expected reward and identify a connection between expected future reward and distance to a goal in a symbolic representation of the safety

component. We then introduce an algorithm for safe and efficient RL using this distance. By reasoning over the specification and a symbolic goal at an abstract level, an additional reward function is derived *automatically* and supplied to the low-level learner, following the tradition of potential-based reward shaping. This ensures that the optimality of the solution is not at stake if the symbolic plan is incomplete or even incorrect. Our algorithm includes an approach to estimate the *shaping rewards* in an online fashion so that no additional hyperparameters are required.

We evaluate the novel approach, called planning for potential (P4P), on a grid world and on a conversational product recommender. The former was inspired by previous work on safe RL whereas the latter contains real-world regulatory constraints from the banking domain. We compare P4P with a 'vanilla' unsafe baseline and a safe baseline. Additionally, we analyze performance of P4P on increasingly constrained problems. We find that P4P scales well with constraint complexity, is robust with regard to its additional parameter and significantly outperforms the baselines in terms of safety and obtained reward.

This chapter is structured as follows: after the preliminaries, we formally introduce the setting of environments with symbolic safety constraints and derive a bound on performance of safe policies. We then show a relation between rewards and reasoning over symbolic constraints using a distance metric. This metric is based on the number of transitions in an automaton representation of the symbolic safety component. We then use this relation in a novel algorithm to improve sample efficiency of reinforcement learning. We test this algorithm on a simple grid world with a tabular RL algorithm and on a realistic conversational product recommendation benchmark with DRL. The proposed algorithm outperforms all baselines and is the only algorithm capable of solving the realistic task, indicating that symbolic reasoning at an abstract level can be combined with learning via reward shaping as proposed in the P4P algorithm.

7.2 Related Work

In this section, we relate this work to the wider body of work on symbolic safety constraints in RL and the use of symbolic reasoning to improve RL agents. Starting with RL under symbolic safety constraints, we group all the works discussed in the introduction. On top of these, [410] encode legality of actions explicitly into rules. [358] specifically target learning normative behaviors in a particular normative framework, whereas we focus on high-level, intensional safety constraints. More importantly, most of these target environments where being safe is positively correlated with high total reward which we show to be a special case of safety-constrained RL. The setting of safe RL under constraints that may impact performance negatively was empirically identified in [184]. We analyze this problem theoretically and propose an algorithm to learn efficiently in this setting. These works are related in the sense that the contributions presented here are complementary: we argue that symbolic reasoning and reward shaping are important components to making these systems

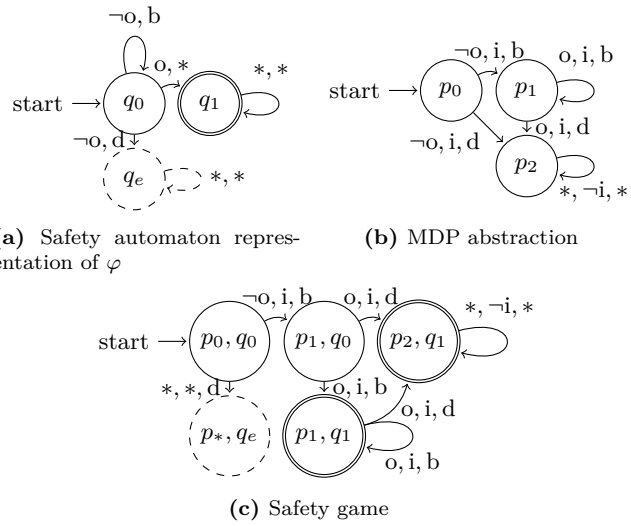
viable in realistic scenarios where rewards and staying safe are not positively correlated.

Closely related are works on restraining bolts by [74, 75]. These use a similar form of reward shaping based on progress in an automaton representation of $\text{LTL}_f/\text{LDL}_f$ specifications of undesired behaviors. These works, however, target a setting in which an external regulator has no control over the agent except for external transitions to the reward. This is overly restricted for cases where the agent is controlled by actors that want to adhere to safety specifications such as in healthcare. More importantly, this setting eliminates guarantees of safety. This work, on the other hand, targets a setting in which we control the agent fully and where provable guarantees are required. Another recent work by [141], presents an approach for safety-constrained RL under the assumptions of knowledge about the transition function, full observability of adjacent state labels and a task fully expressed in LTL. Our work only requires prior knowledge of nonzero transition probabilities at a symbolic level and a goal that expresses which parts of the state-action space are associated with high reward.

A second line of related work aims to inform a learner of knowledge obtained by symbolic reasoning or planning. [134] combine STRIPS-based plans with reward shaping. More recently, several works have proposed using some normal form for representing reward functions [37, 48, 113, 157]. Of specific interest is the work proposing to specify the reward function as an LTL formula and derive intermediate rewards for reward shaping [47]. These works focus on a scenario where the full task can be represented as an LTL goal whereas our approach targets unknown numeric reward function to which a reward is added. We, on the other hand, combine sub-symbolic learning with high-level symbolic reasoning to improve efficiency in a setting with unknown MDP reward and transition functions.

A recent work by Hasanbeig et al. [142] proposes various interesting innovations in this setting. The most relevant of these in relation to our work is an intrinsic reward based on the observation of novel state labels. This intrinsic reward differs from our shaping reward in four important ways. Firstly, it is defined over the state labeling vocabulary Σ_I whereas our approach is based on automata states and hence captures information over *traces* of *both* state and action labels, i.e. over $(\Sigma_I \times \Sigma_O)^\infty$. Secondly, the intrinsic rewards proposed by Hasanbeig et al. would also reinforce moving further from the goal if there happen to be novel labels there. Our shaping approach only reinforces getting closer to the goal. Thirdly, the intrinsic rewards of Hasanbeig et al. do not rely on potentials and hence may produce optimal policies that are suboptimal to original reward function [25, 44]. Finally, we contribute an approach for tuning the single novel hyperparameter in our approach automatically and on-the-fly.

Illanes et al. [160] inform an RL agent with high-level symbolic instructions using the options framework in which low-level learned policies can be reused. The instructions are of a directive nature which is arguably less generic than the constraints used here. Additionally, policies learned in the option framework are sub-optimal whereas our reward shaping approach maintains optimality



(a) Safety automaton representation of φ

(b) MDP abstraction

(c) Safety game

Figure 7.1: Example of automata for a car exiting a gated parking lot. The agent can either **d**rive or push a **b**utton next to the gate, i.e. $AP_O : \{\text{d, b}\}$. State labels indicate whether the gate is **o**pen and whether the car is **i**n the parking lot, i.e. $AP_I : \{\text{o, i}\}$. (a) An automaton representation of $\varphi : \neg\text{d}\mathbf{W}\text{o}$ to express that the gate should be open before the car may drive (b) MDP abstraction with all transitions with nonzero probability in the underlying MDP. Initially, the gate is not open and the car is inside. Error transitions and state are not included for legibility. (c) The result of combining (a) and (b) to form a safety game (excluding abstraction errors). Transitions marked with a solid line are part of the safe strategy. Action 'd' in (p_0, q_0) is not part of the safe strategy since the resulting state is an error state.

guarantees of the underlying learner.

7.3 Preliminaries

7.3.1 Safety Specifications and Shield Synthesis

We define a finite or infinite sequence of elements from some alphabet as a word and a linear-time (LT) property as a set of finite or infinite words over the alphabet $\Sigma : 2^{AP}$, where AP a set of atomic propositions. We focus on safety properties in terms of system input and output and identify subsets of AP and Σ, relating to these as AP_I, Σ_I for inputs and AP_O, Σ_O for outputs. An invariant is an LT property that has to hold in all reachable states for some system, for example "a product may only be recommended if it matches the customer risk profile". Safety properties generalize invariance properties to include patterns over time, for example "products may only be recommended *after* the customer's objectives are known" [19].

Safety properties can be expressed in a formal language that extends pro-

positional logic with temporal operators. Linear temporal logic (LTL) is such a logic [275]. LTL extends propositional logic with temporal modal operators **X** (next) and **U** (until). $\mathbf{X}\varphi$ expresses that a formula φ must be true the next time step and $\psi\mathbf{U}\varphi$ expresses that ψ has to hold at least until φ becomes true. From these, the operators $\mathbf{G}\varphi$ (globally) and $\mathbf{F}\varphi$ (finally) can be defined to express that, from a particular step onward, φ has to respectively hold always and at some point in the future respectively. Additionally, the operator **W** can be derived, which 'weakens' the **U** operators assumptions by allowing that its right-hand-side may or may not be the true in the future.

A LTL safety specification φ_s can automatically be converted into an automaton that represents it. A deterministic finite automaton (DFA) $\varphi_a = \langle \mathbb{Q}, q_0, \Sigma, \delta, \ \rangle$ consists of a set of states $\mathbb{Q}$, an initial state $q_0 \in \mathbb{Q}$, an alphabet $\Sigma = \Sigma_I \times \Sigma_O$, a transition function $\delta : \mathbb{Q} \times \Sigma \to \mathbb{Q}$ and a set of safe state $\mathbb{F} \subseteq \mathbb{Q}$. A *run* is a finite or infinite sequence of states $\bar{q} = q_0, q_1, \cdots \in \mathbb{Q}^\infty$ induced by a trace $\bar{\sigma} = \sigma_0, \sigma_1, \ldots, \in \Sigma^\infty$ of some system such that $\forall i \in \mathbb{N}, q_{i+1} = \delta(q_i, \sigma_i)$. A trace $\bar{\sigma}$ of some system satisfies specification φ_s and its representation φ_a iff the corresponding run $\bar{q}$ visits safe states only, i.e. $\forall i \in \mathbb{N}, q_i \in \mathbb{F}$. An example LTL specification and its automaton representation can be found in Figure 7.1a.

If a model of the environment is available, a reactive system that always produces output in accordance with a specification φ_s can be generated. This challenging task is known as *reactive synthesis* [276]. A typical strategy is to formulate the problem as a two-player alternating game between the system and an adversarial environment. Such a safety game can be expressed as a tuple $\mathcal{G} : \langle \mathbb{G}, g_0, \Sigma_I, \Sigma_O, \delta, \mathbb{F} \rangle$ with a finite set of game states $\mathbb{G}$, initial state $g_0 \in \mathbb{G}$, a transition function $\delta : \mathbb{G} \times \Sigma_I \times \Sigma_O \to \mathbb{G}$ and a set of safe states $\mathbb{F} \subseteq \mathbb{G}$. During the game and for the current state $g \in \mathbb{G}$, the environment first chooses some $\sigma_I \in \Sigma_I$ after which the system chooses $\sigma_O \in \Sigma_O$ and the game transitions to state $g' = \delta(g, \sigma_I, \sigma_O)$. The resulting (infinite) sequence $\bar{g} = g_0, g_1, \ldots$ is called a play and is won by the system iff all visited states are safe: $\forall g_i \in \bar{g}, g_i \in \mathbb{F}$. A winning memoryless strategy is a function $\rho : \mathbb{G} \times \Sigma_I \to \Sigma_O$ if all plays $\bar{g}$ that can be constructed using it are won by the system. Standard algorithms can compute such a winning strategy if it exists [229].

Shield synthesis is a particular kind of reactive synthesis in which an existing system is assumed and in which an external component to correct the system output is computed. The correction is guaranteed to change the output of the original system so that it satisfies some specification with minimal interference given an abstraction of the system [34]. An abstraction of the system describes how its executions can possibly evolve, and provides the needed information about the environment to allow planning ahead w.r.t. the safety properties of interest. The model required is typically of limited size as result. More so, as it can be expressed in an equivalent lifted representation. It may therefore be easy to construct or learn from data. An illustrative abstraction of an MDP and resulting safety game can be found in Figure 7.1. The corresponding shield would replace any unsafe action with a next best safe action.

7.3.2 Reinforcement Learning

RL provides a framework for selecting actions in an environment in order to collect a maximum number of rewards over time [338, 386]. RL deals with problems formalized as Markov decision problems (MDP). We here define a MDP as a tuple $M : \langle S, A, T, R, \gamma, S_0 \rangle$ where $S \in \{s^{(1)}, \ldots, s^{(n)}\}$ is a finite set of environment states, $A \in \{a^{(1)}, \ldots, a^{(m)}\}$ a finite set of agent actions, $T : S \times A \times S \to [0, 1]$ a probabilistic transition function, $R : S \times A \times S \to [R_{\min}, R_{\max}]$ a reward function with $R_{\min}, R_{\max} \in \mathbb{R}$, $\gamma \in [0, 1)$ a discount factor to balance current and future rewards and S_0 a distribution of initial states: $s_0 \sim S_0$. The agent observes an environment state s_t at each time step t and performs some action a_t up to some end time $\mathcal{T}$, following some policy $\pi \in \Pi : S \times A \to [0, 1]$ and collects reward $r_t = R(s_t, a_t)$.

If some expectation $\mathbb{E}_\pi$ can be formulated to express the sum of rewards by following some π, then values V and Q can be assigned to a state s and a tuple (s, a) respectively for that π:

$$V_\pi(s) = \mathbb{E}_\pi \left[\sum_{k=t}^{\mathcal{T}=\infty} \gamma^{k-t} r_k | s_t = s \right] \tag{7.1}$$

$$Q_\pi(s, a) = \mathbb{E}_\pi \left[\sum_{k=t}^{\mathcal{T}=\infty} \gamma^{k-t} r_k | s_t = s, a_t = a \right] \tag{7.2}$$

A policy π is the optimal policy π^* if it results in the highest obtainable reward: $\forall s \in S, \forall \pi \in \Pi, \forall a \in A : Q_{\pi^*}(s, a) \geq Q_\pi(s, a)$. Finding π^* can be addressed by conditioning the policy on a set of parameters $\pi(s_t|\theta) = a_t$ and finding parameter values θ^* that maximize the corresponding reward by a learning algorithm. For example, θ can be weights of a neural network updated with gradient descent.

A particularly popular parameterized approach of learning an approximation of π^* is known as deep Q-Networks (DQN) [238, 240]. DQN uses a neural network with weights θ to predict $Q_\pi(s, a|\theta)$ and selects actions uniform randomly with some probability $\epsilon \in (0, 1]$ or greedily with respect to $Q_\pi(s, a)$ with some probability $1 - \epsilon$ at each step t. The resulting tuple (s_t, a_t, r_t, s_{t+1}) is added to a buffer or data set D as (s, a, r, s'). Weights are updated in iterations. For every iteration i, the current weights θ_i are updated to minimize the loss function $\mathcal{L}_i(\theta_i) =$

$$\mathbb{E}_{(s,a,r,s') \sim U(D)} \left(r + \gamma \max_{a'} Q(s', a'|\theta_i^-) - Q(s, a|\theta_i) \right)^2 \tag{7.3}$$

where $U(D)$ is a uniform random sample of D and θ_i^- the parameters used in action selection during iteration i. These parameters θ_i^- are only replaced with θ_i every C iterations and held fixed otherwise as this increases stability of the learned Q-network over time, hence improving performance.

7.4 Safe Reinforcement Learning

RL has proven capable of learning complex behaviors from interactions with an environment in a trial-and-error fashion alone. Symbolic reasoning, on the other hand, is well suited when safety guarantees on behavior are necessary. Safe RL combines these in order to learn complex behaviors under strong guarantees of safety, both during and after learning. In this section, we introduce safety-constrained environments following [5], identify that safe policies are not expected to outperform unsafe policies and then analyze how safety constraints impact the expected future reward.

Definition 7.1 (Safety-constrained environments)**.** A *safety-constrained environment* is a tuple $E : \langle M, AP, L_I, L_O, \varphi \rangle$ where $M : \langle S, A, T, R, \gamma \rangle$ is an MDP, φ is a safety specification with propositions $AP : AP_I \cup AP_O$ and labelling functions $L_I : S \to 2^{AP_I}$, $L_O : A \to 2^{AP_O}$.

Definition 7.2 (Safe policies)**.** A policy $\pi \in \Pi : S \times A \to [0, 1]$ is *safe* in E if for any sequence $s_t, a_t, s_{t+1}, a_{t+1}, \dots$ it generates with nonzero probability, the corresponding sequence of labels $(L_I(s_t) \cup L_O(a_t), L_I(s_{t+1}) \cup L_O(a_{t+1}), \dots)$ satisfies φ. The set of safe policies in E is denoted Π_E.

For the purposes of this chapter, the MDP transition function T is unknown. If results of actions are unknown, safety of a given policy cannot be verified upfront without further assumptions. In order to ensure safety, however, we need only know which sequences of labels for a given policy in an environment have a nonzero probability. These can be modeled with an automaton abstraction of the MDP.

Definition 7.3 (MDP abstractions)**.** Given an environment E, the automaton $\varphi_M : \langle \mathbb{Q}, q_0, \Sigma_I \times \Sigma_O, \delta, \mathbb{F} \rangle$ is an abstraction of M if for every trace $s_0, s_1, \dots \in S^\infty$ and corresponding action sequence $a_0, a_1, \dots \in A^\infty$ with nonzero probability in E, for every run $\bar{q} : q_0, q_1, \dots \in \mathbb{Q}^\infty$ with $q_{i+1} = \delta(q_i, L_I(s_i), L_O(a_i))$, this run $\bar{q}$ visits only states in $\mathbb{F}$.

Remark 7.1. It can be verified whether an automaton is an abstraction of M. If a run is generated in which some state $q_i \notin \mathbb{F}$ then it is not an abstraction of M. This property can be used to refine the abstraction when it is tested or to hand over control to a human operator or fallback policy.

An abstraction can be used to synthesize safe policies. The cross product of the abstraction and an automaton representation of the specification forms a safety game from which a safe strategy can be computed as described in the previous section. Policies following this strategy are provably safe in E [5]. We now introduce a performance bound for safe policies.

Theorem 7.1 (Performance bound)**.** *For any environment E with MDP M and safe policies Π_E, let $\pi^*_E \in \Pi_E$ be the optimal safe policy and $\pi^*_M \in \Pi$ be the optimal (possibly unsafe) policy. Then $\pi^*_E \leq \pi^*_M$ where $\pi_1 \leq \pi_2$ iff $\forall s \in S, \forall a \in A, Q_{\pi_1}(s, a) \leq Q_{\pi_2}(s, a)$.*

Proof. $\Pi_E \subseteq \Pi$, hence $\pi^*_E \in \Pi$ and $\pi^*_E \leq \pi^*_M$. □

Theorem 7.1 shows that safe policies are generally not expected to outperform their unsafe counterparts in terms of reward: the environments targeted in previous work where being unsafe leads to low rewards are a special case of safety constrained environments. We continue to investigate when constraints negatively impact expected reward. In order to do so, we first introduce the notion of a goal. Problems with pre-specified goals are approached within the framework of goal-based MDPs in RL. Such MDPs terminate if a goal state $s_g \in S$ is reached and have a reward function of the form $R(s, a, s') = 1$ if $s' = s_g$ and 0 otherwise. Here, we consider goal-based problems within the framework of safety-constrained environments and define them in terms of AP.

Definition 7.4 (Goals)**.** For a safety-constrained environment $E : \langle M, AP, L_I, L_O, \varphi \rangle$, a goal $\sigma_g \in 2^{AP}$ is reached if an action $a \in A$ with labeling $\sigma_a : L_O(a)$ is selected in a state $s \in S$ with labeling $\sigma_s : L_I(s)$ such that $\sigma_g \subseteq \sigma_s \cup \sigma_a$.

Definition 7.5 (Goal-based environments)**.** An environment $E : \langle M, AP, L_I, L_O, \varphi, \sigma_g \rangle$ with goal σ_g is goal-based if it has a reward function of the following restricted form: $R(s, a, s') = 1$ if the goal is reached by performing a in s and $R(s, a, s') = 0$ otherwise.

How a constraint specifically impacts expected reward depends on the particular goal, the constraint and the transition function T, which is unknown in the case of interest here. Even in this case, however, the impact of a constraint can be derived in some illustrative cases which serve as an inspiration to the algorithm presented later. First, we focus on the case where the goal can be reached immediately.

Theorem 7.2 (Q values and reaching goals)**.** *For any goal-based environment E with safe optimal policy π^*_E and for any $s, s' \in S$ and any $a, a' \in A$ with $\sigma_s : L_I(s), \sigma_{s'} : L_I(s'), \sigma_a : L_O(a)$ and $\sigma_{a'} : L'_O(a')$:*

$$\begin{aligned} Q_{\pi^*_E}(s, a) > Q_{\pi^*_E}(s', a') \quad & \textit{if} \\ \sigma_g \subseteq \sigma_s \cup \sigma_a \quad & \textit{and} \\ \sigma_g \not\subseteq \sigma_{s'} \cup \sigma_{a'} \quad & \end{aligned}$$

Proof. If $\sigma_g \subseteq \sigma_s \cup \sigma_a$ then performing a in s ends the episode in E and yields the maximum obtainable reward of $R(s, a, \cdot) = 1$. Since $\sigma_g \not\subseteq \sigma_{s'} \cup \sigma_{a'}$, $R(s', a', \cdot) = 0$ and therefore $Q_{\pi^*_E}(s, a) > Q_{\pi^*_E}(s', a')$. □

Corollary 7.3. *For an environment E in some state $s \in S$ for any two actions $a, a' \in A$ such that $\sigma_g \subseteq \sigma_s \cup \sigma_a$ and $\sigma_g \not\subseteq \sigma_s \cup \sigma_{a'}$:*

$$Q_{\pi^*_E}(s, a) > Q_{\pi^*_E}(s, a')$$

Proof. By substituting $s' = s$ in Theorem 7.2. □

When the goal cannot be reached immediately, reaching the goal requires multiple transitions in both the underlying MDP and the safety game $\mathcal{G}$. The minimum number of transitions required in $\mathcal{G}$ for any of its safe states $f \in \mathbb{F}$ is denoted $\Delta_{\mathcal{G}}(f_i, \sigma_g)$. It can be derived using planning and interpreted as the distance from f to the goal while staying safe.

Definition 7.6 (Distance in safety games)**.** For a goal-based environment E with safety game $\mathcal{G} : \langle \mathbb{Q}, q_0, \Sigma_I, \Sigma_O, \delta, \mathbb{F} \rangle$ with a winning strategy ρ, a distance map $\Delta_{\mathcal{G}} : \mathbb{F} \to \mathbb{N}_1$ is defined as the length of the shortest play $\bar{f} = f_i, \ldots, f_k, f_l$ starting in any $f_i \in \mathbb{F}$ with $\delta(f_k, \sigma_i, \sigma_o) = f_l$ such that the play $\bar{f}$ can be constructed with ρ and $\sigma_g \subseteq \sigma_i \cup \sigma_o$.

Remark 7.2. $\Delta_{\mathcal{G}}(f, \sigma_g) = 1$ iff $\sigma_g \subseteq \sigma_s \cup \sigma_a$ for some $a \in A$ in a given $s \in S, f \in \mathbb{F}$

Reaching the goal may also require multiple transitions in the MDP. Since T is unknown in the setting of interest, the expected number of transitions is unknown as well. Therefore, we look into a second illustrative case where safety constraints and goals are defined fully on the action space, i.e. $AP_I = \emptyset$.

Theorem 7.4 (Q values and distances)**.** *For any goal-based environment E with safety game $\mathcal{G}$ and for any $s, s' \in S$, $a, a' \in A$, $f, f' \in \mathbb{F}$:*

$$\begin{aligned} Q_{\pi^*_E}(s, a) &> Q_{\pi^*_E}(s', a') && \textit{if} \\ \Delta_{\mathcal{G}}(f', \sigma_g) &> 1 && \textit{and} \\ \Delta_{\mathcal{G}}(\delta(f, \emptyset, \sigma_a), \sigma_g) &< \Delta_{\mathcal{G}}(\delta(f', \emptyset, \sigma_{a'}), \sigma_g) \end{aligned}$$

Proof sketch. Suppose that the consequent of the equation holds. Let n denote the distance towards a goal by taking a, $n = \Delta_{\mathcal{G}}(\delta(f, \emptyset, \sigma_a), \sigma_g)$, and n' denote the distance towards a goal by taking action a', $n' = \Delta_{\mathcal{G}}(\delta(f', \emptyset, \sigma_{a'}), \sigma_g)$. An optimal safe policy in E takes n time steps to reach σ_g after performing a and similarly so for n' and a'. Now by substituting $\mathcal{T} = n$ and $\mathcal{T} = n'$ in Equation 7.2 and since $\gamma < 1$ and $n < n'$ we find $Q_{\pi^*_E}(s, a) > Q_{\pi^*_E}(s', a')$. □

Remark 7.3. An inequality for a single $s \in S$ and single $f \in \mathbb{F}$ can be derived analogously to Corollary 7.3.

The presented analysis shows how Q values relate to goals and distances to goals in safety games. Although the analysis is targeted at goal-based environments, their implications are also applicable to other settings, such as those in which the symbolic goal serves as a proxy for a high reward area of the state-action space. Our analysis indicates that there are two classes of safety constrained environments and it shows how to identify them. In the first class, safe policies are expected to perform equally to unsafe policies in terms of obtained rewards. The distance from initial state to a goal in the associated safety game is not impacted by the safety constraints in these environments: they would be equal to the distance of a safety game resulting from vacuous constraints that always hold. In the second class of safety constrained environments, unsafe policies are expected to outperform safe ones. The safety

Algorithm 4 Efficient RL in a safety-constrained environment with P4P.

Input: specification φ, abstraction φ_M, goal σ_g
Parameters: cost $c > 0$
Output: policy π

1: $\mathcal{G} \leftarrow \varphi \times \varphi_M$
2: $shield \leftarrow$ computeShield($\mathcal{G}$)
3: $\Phi \leftarrow$ potentials($\mathcal{G}, \sigma_g, c$) {see Algorithm 5}
4: Initialize π arbitrarily
5: **for all** episode **do**
6: $g \leftarrow g_0$ from $\mathcal{G}$
7: **for all** time step **do**
8: $s \leftarrow$ get from environment
9: $\sigma_s \leftarrow L_I(s)$
10: $a \leftarrow$ select safe action from $shield, \pi$
11: take action a
12: $r, s' \leftarrow$ get from environment
13: $\sigma_a \leftarrow L_O(a)$
14: $g' \leftarrow \delta(g, \sigma_s, \sigma_a)$
15: $r' \leftarrow r + \gamma\Phi(g') - \Phi(g)$
16: Update π using (s, a, s', r')
17: $s \leftarrow s', g \leftarrow g'$
18: **end for**
19: **end for**
20: **return** π

constraints add transitions to the shortest path towards a goal. As constraints are added, the distance from the goal grows. Every transition that a safety constraint contributes, leads to at least one additional transition for the learner to incorporate and as such makes the learning problem more complex. The next section introduces an algorithm to improve the scalability of safe RL as problems become more constrained.

7.5 Planning for Potential

In this section, we propose a scalable and efficient RL algorithm adhering to a safety specification. The algorithm uses symbolic knowledge available in a safety-constrained environments to speed up the learning. Optimality is preserved in cases of incomplete or even incorrect prior knowledge. In the proposed approach, the learner is informed of progress with respect to a symbolic goal by transforming the reward function *automatically*. The approach follows the tradition of potential-based reward shaping which we introduce first.

7.5.1 Reward shaping

Shaping is a technique within RL in which the original MDP $M : \langle S, A, T, R, \gamma \rangle$ is replaced with a surrogate $M' : \langle S, A, T, R', \gamma \rangle$ in order to guide the learner. It is desirable that R' is easier to learn but yields only optimal policies that are also optimal under the original R. Ng, Harada and Russell [254] showed that $R' = R + \mathcal{S}$ with shaping function $\mathcal{S} : S \times A \times S \to \mathbb{R}$ such that $\mathcal{S}(s, a, s') = \gamma\Phi(s') - \Phi(s)$ with so-called potential $\Phi : S \to \mathbb{R}$ are the only R' that guarantee that any policy optimal in M' is also optimal in M if no further information on transition and reward functions is known. The challenge now consists of defining a potential function Φ that informs the learner.

In safety-constrained RL, knowledge about the task is available in symbolic form in the safety component. To use this knowledge, a description of state-action tuples associated with high reward are described in symbolic form. The distance to this symbolic goal is established with planning. By comparing distances prior to and after taking an action, we know whether that action contributed to reaching the symbolic goal. The agent is informed of this with potential-based reward shaping, hence we call this approach planning for potential (abbreviated P4P).

7.5.2 Algorithm

P4P is listed in Algorithm 4. For a given safety constraint and MDP abstraction, a safety game and shield are computed following Alshiekh et al. [5]. Next, a map of potentials Φ is computed for all safe states of the safety game, after which the learning loop begins. This is a traditional RL learning loop with two modifications: actions are selected from the set of safe actions (line 11) and the reward is augmented with the difference between potentials (line 15).

Algorithm 5 lists how the map of potentials Φ can be computed. The minimum number of transitions, or *shortest distance*, from each state to the goal state are determined using symbolic planning. These distances can be derived prior to interacting with the environment and with minimal knowledge of the MDP transition function. This makes them well suited as a signal of progress for an exploring agent in a safety-constrained environment. The algorithm presented here computes potentials for all states upfront. Although limited in terms of scalability with respect to the safety game state space, this simple approach will suffice for many settings for two reasons. Firstly, the safety game only includes aspects of safety and this 'abstraction' over the full MDP yields relatively small safety games. Secondly, the calculation of these values is a one-time operation that is easily dwarfed by the iterative training approach of many RL algorithms used in practice. If necessary, however, more elaborate methods can be applied. Any solution to the unweighted single start shortest path algorithm tailored to the desired performance characteristics can be used. More so, the distances and potentials can be calculated in an online fashion, i.e. deferring the calculation of these values for all states to the moment of first visiting them to increase scalability for settings with a large number of safety

Algorithm 5 Planning-based potentials Φ.

Input: safety game $\mathcal{G}$, goal σ_g
Parameters: cost $c > 0$
Output: potentials Φ

```
1: Initialize Φ for all states in 𝔽
2: Initialize dist for all states in 𝔽
3: for all state ∈ 𝔽 do
4:    dist(state) ← CalcDist(state)
5: end for
6: return Φ
7: procedure CalcDist(state)
8:    minDist ← ∞
9:    for all σ_s, σ_a ∈ Σ_I, Σ_O do
10:      if σ_g ⊆ σ_s ∪ σ_a then
11:         return 1
12:      end if
13:      next ← δ(state, σ_s, σ_a)
14:      if next ∈ 𝔽 then
15:         dist ← CalcDist(next)+1
16:         minDist ← min(dist, minDist)
17:      end if
18:   end for
19:   return minDist
20: end procedure
```

game states that are not visited in practice.

In order to convert distances to progress, the difference in distances to a goal between the initial state and any other state are calculated. This difference represents the progress for any given state. It is multiplied with cost parameter c to offset the 'costs' already incurred while moving closer to the goal. A reward bonus is given to state-action pairs closer to the symbolic goal in accordance to the inequalities of Theorems 7.2 and 7.4. As P4P uses potentials to augment the obtained rewards, there are no formal requirements on c except for being > 0 by Theorems 7.2 and 7.4. In the case of stochasticity in updates of estimates for V and Q, however, the value of c may affect convergence. In this case, c can be set using knowledge of the domain, tuned as a hyperparameter or estimated during learning in an online fashion, see Section 7.5.4.

7.5.3 Example

For the safety-constrained environment in Figure 7.1, the goal is to not be in the parking lot $\sigma_g : \neg\mathrm{i}$. This goal can be reached safely by taking any action in (p_2, q_1). We can only visit this state by first visiting (p_1, q_0) which has potential:

$$\begin{aligned}\Phi((p_1 q_0)) &= c * (\Delta_{\mathcal{G}}((p_0, q_0)) - \Delta_{\mathcal{G}}((p_1, q_0)) \\ &= c * (3 - 2) = c\end{aligned}$$

Transitioning from $(p_0, q_0) \to (p_1, q_0)$ therefore yields an additional reward of $\gamma \cdot c$. On the other hand, the additional reward for transitioning from $(p_1, q_0) \to (p_1, q_1) = \gamma \cdot c - c$ and effectively reflects the fact that visiting (p_1, q_1) was not necessary in order to reach the goal.

The additional rewards supplied to the learning algorithm guide the learner to prefer certain states and actions. Doing so involves balancing the provided bias to increase learning without limiting the learner. Two mechanisms in P4P ensure that the amount of bias is suitable. Firstly, the use of potential-based shaping ensures that the optimal policy under transformed reward function is optimal under the original reward as well (see Section 7.5.1). Secondly, progress in the algorithm is based on the shortest safe path toward the goal in the safety game. This may be overly optimistic when the trajectories in the MDP corresponding with this path have low probability. If this is the case, the learner will observe these trajectories infrequently and the effects of P4P will be limited. P4P thus successfully leverages all information available at the symbolic level without overly biasing the learner.

In P4P, distances to a goal are derived by symbolic reasoning and subsequently used to inform a learner via reward shaping. The shaped reward function is more dense than the original reward function and guides the agent towards promising regions of the state-action space in exploration phases. Furthermore, rewards are obtained as the agent progresses towards its goal. Thus, a part of the value assignment problem of RL is already solved for the learner. Finally, the usage of potential-based shaping guarantees that optimality guar-

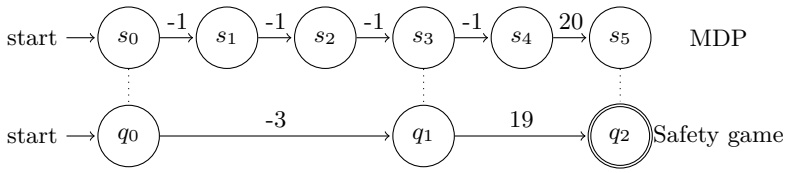


Figure 7.2: Traces for a successful episode in a hypothetical MDP (top) and safety game (bottom). Transition labels indicate rewards associated with that transition.

antees for the underlying learning algorithm also apply to P4P, even if the provided goal is incomplete or incorrect.

7.5.4 Estimation of c

The parameter c in Algorithms 4 and 5 controls the additional rewards given to the agent for each transition towards a goal in the safety game. As previously noted, there are no formal requirements on setting parameter c, except for $c > 0$ according to Theorems 7.2 and 7.4. Although convergence towards the optimal policy is not affected by the value of c in the long run, a suitable value can impact speedups obtained when using function approximation such as in DRL. In this section, we describe two ways to find a suitable value for c. They can be used depending on the available upfront knowledge. The first approach is based on the size of the safety game and existing knowledge of the maximum obtainable reward. If this knowledge is available, then parameter c can be set using the following heuristic:

$$c := \frac{\hat{R}_{\max}}{dist(g_0)} \tag{7.4}$$

where $\hat{R}_{\max}$ denotes an estimate of $R_{\max}$, the maximum of R, and $dist(g_0)$ denotes the distance of the initial state in the safety game to its closest goal as calculated in Algorithm 5.

Alternatively, c can be tuned in an online fashion from interactions alone. In order to do so, rewards are to be associated with transitions in the safety game. Figure 7.2 shows how state transitions in the MDP are associated with state transitions in the safety game. Although the agent is successful in the end, a reward of -3 is incurred by transitioning $q_0 \to q_1$. These 'costs' for safety game transitions are stored in a buffer, averaged and multiplied with -1 in order to establish c in an online fashion. In some environments, reward is 0 most of the time. An example environment is the goal-based environment in Definition 7.5. In such environments, no rewards are obtained by transitioning in the safety game. An elegant solution for this problem is to linearly transform all observed rewards to $\mathbb{R}_{\leq 0}$ as a first step of estimating c dynamically.

7.6 Experimental Setup

The experimental setup was designed to answer four specific research questions: how can RL agents learn safely and efficiently (Q1)? How do different constraints impact efficiency (Q2)? How sensitive is the proposed approach to its hyperparameters (Q3)? In order to answer these questions, the proposed P4P approach was evaluated in two environments. In these environments, agents were trained in different conditions: a baseline not adhering to the specification ('unsafe'), a 'shielded' baseline from Alshiekh et al. [5] and three P4P variants. In the first of these variants, the P4P hyperparameter c is estimated online as proposed in Section 7.5.4. The other two variants use a fixed value for this parameter in order to answer Q3. The values are set to overestimate ('P4P-o') and underestimate ('P4P-u') the true cost.

7.6.1 Grid world environment

A grid world environment from [5] with an 'exact' abstraction was used. Grid world environments are often used in RL as they are relatively simple to grasp but include many of the characteristics of more challenging learning problems and are useful to e.g. test intuitions. In the grid world used here (Figure 7.3a), being safe does not correlate with high reward. A reward of 1 is obtained when all regions have been visited in order and 0 otherwise. States in gray can be visited but are to be avoided according to the safety requirements. The goal was formulated as $area1 \wedge area2 \wedge area3 \wedge area4$. Parameter c was estimated in an online fashion by storing rewards obtained for each transition in the safety game (see Section 7.5.4). Additionally, we ran experiments with over- and underestimates for c to answer Q3. These were established as follows. In this environment, the average cost of a safety game transition can be calculated. Based on the number of actions necessary to visit all regions safely and the reward that is obtained, the average cost was established at 8e−3. The cost parameter was set to $c = 1e{-}5$ as an underestimate (P4P-u) and $c = 2$ as an overestimate (P4P-o) in order to test robustness to this parameter. All agents were trained using ϵ-greedy tabular Q-learning with $\alpha = .2$ and $\gamma = .95$ [384]. Exploration parameter ϵ was cooled down linearly from .2 to .01 over the total number of 1e4 episodes. Episode length and number of violations of the safety specification were recorded across ten random seeds.

7.6.2 Conversational Recommendation Environment

sd A realistic and high-dimensional environment from the banking domain was included due to the availability of real-world constraints from Chapter 3. In this conversational recommendation environment, the agent interacts with a simulated user. The task is to recommend a product and provide the desired information in a minimal number of turns. Reward is specified as follows: each conversation turn yields a reward of -1 and at the end of each conversation, an additional reward of 20 is obtained if the provided information meets the

	Prop.	Explanation
AP_O	*rec*	The agent makes a recommendation.
	e	The agent explains the expected result of a recommended product.
	ena	The agent explains the need for analysis of the customer profile.
	dsp	The agent discloses the customer profile.
	dvp	The agent recommends a product that deviates from the risk profile.
AP_I	*ok*	The objective of the customer is known.
	cdvp	The customer confirms they want to deviate from their risk profile.
	vp	The customer verifies the risk profile disclosed by the agent.
	ue	The customer indicates to understand the explanation of the result.

Table 7.1: Atomic propositions in the recommender environment.

$\varphi_\#$	Regulatory statement	Specification
1	Explain the expected result, check whether it is understood	$\mathbf{G}(rec \rightarrow ((e \vee rec)\ \mathbf{W}\ ue))$
2	No recommendation if the profile has been disclosed but not verified	$\mathbf{G}(dsp \rightarrow (\neg rec\ \mathbf{W}\ vp))$
3	No deviation from risk profile until customer confirms deviation	$\neg dvp \mathbf{W}\ cdvp$
4	No recommendation until customer objective known	$\neg rec\ \mathbf{W}\ ok$
5	No recommendation until the customer profile has been disclosed	$\neg rec \mathbf{W}\ dsp$
6	No recommendation until the need for analysis has been explained	$\neg rec\ \mathbf{W}\ ena$

Table 7.2: Formalization of regulatory safety statements into LTL specifications.

information need and 0 otherwise. The unconstrained action space consists of 38 actions. States are each represented by a boolean vector with length 136 that describes beliefs over the customers preferences and the dialogue history. The dimensionality of the state-action space is 38×2^{136} and makes this a challenging problem for which function approximation is necessary.

Realistic constraints were constructed from a real-world regulatory document . All statements pertaining to the interaction between a bank and customers were extracted from this document and formalized in consultation with two domain experts. The vocabulary used in the specification is listed in Table 7.1 and the specifications are listed in Table 7.2. Separate specifications were used to gauge the impact of different constraints (Q2). Accuracy was recorded for random rollouts for each separate constraint as a proxy for the 'difficulty' of the constraint. Constraints are presented in decreasing difficulty in Table 7.2. Increasingly difficult specifications were created by combining separate specifications: $\varphi_{1,2} = \varphi_1 \wedge \varphi_2, \varphi_{1,2,3} = \varphi_{1,2} \wedge \varphi_3, \ldots, \varphi_{1-6} = \bigwedge \varphi_{1,\ldots 6}$. For all constraints, the goal *rec* was used.

7

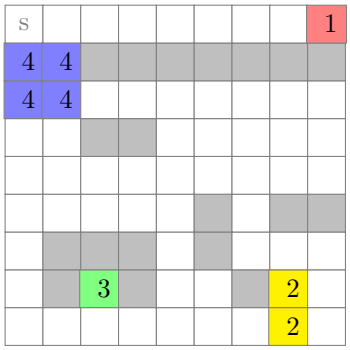
Agents were trained in eight conditions: a baseline, P4P with online estimated c (P4P), P4P with an overestimate of c (P4P-o) and P4P with an underestimate of c (P4P-u), all for both the unsafe and shielded case. The under- and over-estimates of c were determined as follows: the reward of a dialogue turn is -1. However, only some turns result in a transition in the safety game. Therefore, 1 is a reasonable underestimate for c. The overestimate was based on the average number of turns used by an unconstrained agent. After training, it requires on average seven turns to complete the task. Therefore, we used $c = 8$ as an overestimate for each *single* transition towards the goal.

For each condition, five agents with different random seeds were trained on 30K dialogues. After every 1K dialogues, performance was measured on 500 test dialogues. Rewards, accuracy, number of dialogue turns and safety specification violations were recorded. All agents use ϵ-greedy DQN where ϵ is linearly cooled down from .3 to $\epsilon_f = .05$ and with a learning rate $\alpha = 1\text{e}{-4}$. These hyperparameters were selected after a grid search on $\epsilon_s \in \{.3, .5, .9\}$, $\epsilon_f \in \{.05, .3\}$ and $\alpha \in \{1\text{e}{-3}, 1\text{e}{-4}\}$.

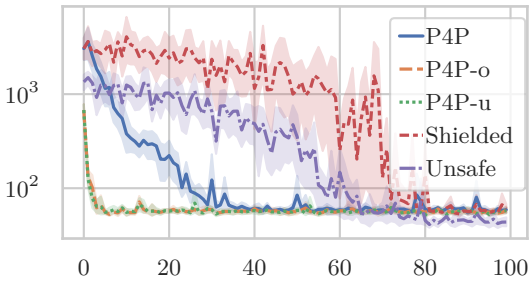
7.7 Experimental Results

7.7.1 Grid world environment

Figure 7.3b shows episode lengths for all included agents in the grid world environment. P4P converges toward an optimal policy quickly (Q1). As a result of the 'exact' abstraction, the potentials reflect progress made for every time step. P4P significantly outperforms both safe and unsafe baselines, which both achieve optimal behavior eventually. More so, we see that both P4P-u and P4P-o converge faster than P4P (Q3). This is explained by the fact that P4P requires a short phase where an appropriate estimation of c is to be learned. Additionally, we see comparable results for P4P-o and P4P-u,



(a) Environment



(b) Episode length per episode in the grid world environment

Figure 7.3: Grid world with start position 's'. Positions marked in gray are to be avoided.

	Unsafe		Safe	
	Baseline	P4P	Shielded	P4P
Reward	5.43 ± 1.94	6.43 ± 1.94	-11.93 ± 1.73	$\mathbf{5.82 \pm 1.96}$
Acc. (%)	65.84 ± 2.75	$\mathbf{79.60 \pm 2.74}$	0.00 ± 0.00	$\mathbf{79.16 \pm 2.74}$
Viol. (%)	80.16 ± 11.36	74.00 ± 7.49	0.00 ± 0.000	0.00 ± 0.000
Turns	7.73 ± 1.539	9.49 ± 1.591	11.93 ± 1.730	10.01 ± 1.561

Table 7.3: Recommendation environment test set results (mean ±95% confidence interval). **Bold** denotes significant improvements w.r.t. baseline/shielded.

which is explained by the relatively small effect of shaping reward scale on the probability of selecting a particular action in the case of ϵ-greedy tabular Q-learning.

7.7.2 Conversational Recommendation environment

Performance metrics for all agents in the conversational recommender environment are listed in Table 7.3. The shielded baseline does not learn to solve the task at hand, i.e. average accuracy is 0.00. In contrast, accuracies for P4P are comparable to the unsafe baseline (Q1). Finally, all unsafe agents violate the specification: rewarding safe behavior is not sufficient for a safe agent. Figure 7.4 shows the accuracy of the tested approaches on varying constraints (Q2). P4P performs comparable to the unsafe baseline and comparable to or better than the safe baseline. Benefits of P4P grow as problems become more constrained (Q2).

We continue to investigate sensitivity to the c parameter by comparing the results between P4P variants (Q3). We first revisit Table 7.3. Both P4P variants converge to high reward policies without a significant difference in reward, accuracy or number of turns between the two variants. However, Figure 7.4

shows differences in data efficiency. Specifically, P4P with $c = 8$ converges to high rewards faster than the $c = 1$ variant. The signal for progress with respect to the safety constraints is more prominent with $c = 8$ without harming overall performance.

7.8 Discussion

This work set out to address the problem of efficient and provably safe RL in settings where being safe need not be associated with high rewards. We formally introduced environments with symbolic safety constraints and showed that the performance of safe policies are only expected to perform equally to unsafe policies in a special case. We analyzed how constraints impact expected future rewards and showed a relation between expected rewards and the progress toward a goal in an automaton representation of the available symbolic knowledge.

We then proposed an algorithm to scale safe RL with constraint complexity based on symbolic reasoning. Reasoning is used to infer progress towards a symbolic goal. A reinforcement learner is then infused with this progress signal using additional rewards, following the convention of potential-based reward shaping. We evaluated the so-called P4P algorithm on two existing environments, one of which with real-world constraints. We found that it significantly outperforms baselines and scales well as problems become more constrained. Additionally, we introduced an approach for tuning its single additional hyperparameter in an online fashion and showed that the algorithm is robust against various values of this parameter.

In P4P, safety constraints are expressed at a symbolic, intensional level and need not align with large total rewards. Such problems are abundant in e.g. regulated domains such as healthcare and finance. Here, regulatory constraints prevent rare yet undesirable events, unwelcome long-term effects and negative externalities. P4P exemplifies that safety in RL can be achieved at negligible performance penalty if learning and reasoning are combined.

The findings presented here inspire various directions for future work. Firstly, we have seen that constraints have a varying effect on learning efficiency. It would be useful if these could be estimated analytically and up-front by building on the framework of safety-constrained environments as first introduced by [5]. Secondly, there is an interesting direction in altering the approach to be applicable to settings that do not necessarily involve safety constraints, but where knowledge about suitable policies is available at a symbolic level. Of particular interest here is the decomposition of the full RL task in subtasks, as has been proposed recently by [9, 160]. The use of constraints may be an interesting alternative if more fine-grained symbolic information is not available. Thirdly, the approach presented here can be combined with methods that learn a set of symbolic labels for the state and action spaces or that learn an automaton representation of the problem [142]. This is of particular interest if the safety constraints are not *strict*, because learning these requires violation

of the constraints during early stages.

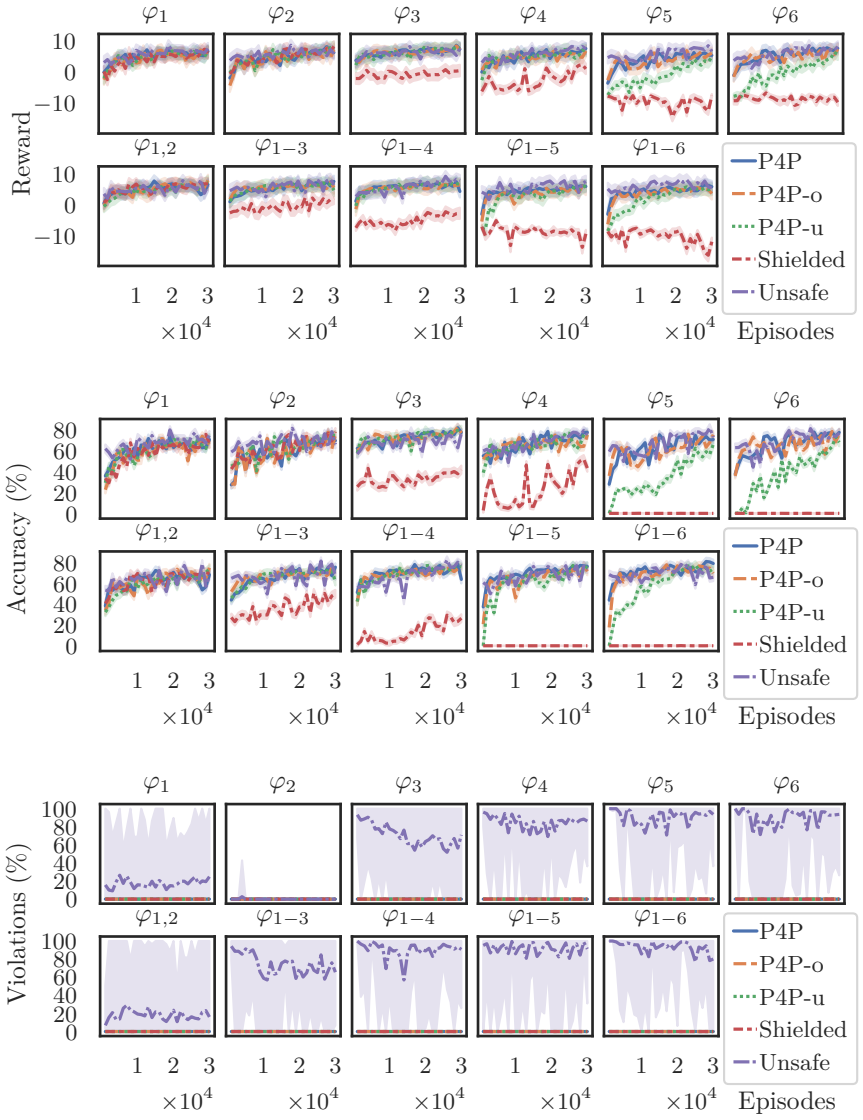


Figure 7.4: Test set results in an increasingly constrained recommendation environment (mean and 95% confidence intervals).

8

Reinforcement Learning with Option Machines

Reinforcement learning (RL) is a powerful framework for learning complex behaviors, but lacks adoption in many settings due to sample size requirements. We introduce a framework for increasing sample efficiency of RL algorithms. Our approach focuses on optimizing environment rewards with high-level instructions. These are modeled as a high-level controller over temporally extended actions known as *options*. These options can be looped, interleaved and partially ordered with a rich language for high-level instructions. Crucially, the instructions may be *underspecified* in the sense that following them does not guarantee high reward in the environment. We present an algorithm for control with these so-called *option machines* (OMs), discuss option selection for the partially ordered case and describe an algorithm for learning with OMs. We compare our approach in zero-shot, single- and multi-task settings in an environment with fully specified and underspecified instructions. We find that OMs perform significantly better than or comparable to the state-of-art in all environments and learning settings.

Based on [P3]:
Floris den Hengst, Vincent François-Lavet, Mark Hoogendoorn, and Frank van Harmelen
Reinforcement Learning with Option Machines
Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22

8.1 Introduction

Reinforcement Learning (RL) is a powerful framework for learning complex behaviors. Sample effiency, however, remains an open challenge in RL and prevents adoption in many real-world settings [86], Chapter 2. Sample efficiency is often improved with knowledge of a good solution, e.g. with demonstrations, increasingly complex tasks [29], intermediate rewards [254] and by decomposing the task into subtasks that are easier to learn [83].

Recently, approaches have become popular for making RL more sample efficient with high-level symbolic knowledge. These methods combine the clear semantics, verifiability and well-understood compositional and computational characteristics of symbolic methods at a high level of abstraction with the power and flexibility of RL at large, low-level action and state spaces [48, 157, 160, 212, 359, 394], Chapter 7. These works demonstrate that symbolic instructions form a compelling complement to RL. A drawback of existing methods, however, is that they require the instructions to fully define the task at hand. Specifically, these assume that high rewards are *always* obtained if the instructions are followed. Such rich instructions, however, may be hard to attain in practice. Firstly, knowledge of a good solution may be tacit. Secondly, the solution space may be so large that only partial instructions are feasible, e.g. chess opening and closing strategies. Finally, the quality of a solution may not be known a priori, e.g. when it depends on the agents' capabilities or user preferences as in Chapter 2.

We therefore target a setting in which an agent is to optimize an environment reward with the help of *underspecified* instructions. These instructions define a solution at a high level of abstraction and, crucially, do not define the task at hand completely: following these instructions does not guarantee a high environment reward. Such instructions are abundant in a vast range of domains, including driving directions and clinical guidelines. In this chapter, we propose and evaluate a framework for sample-efficient RL with underspecified instructions.

The framework consists of a high-level controller over a set of temporally extended actions known as *options* [335] and uses a formalism that allows for looping, interleaving and partial ordering of such options. The policies for these options are trained to optimize an environment return and can be reused both within a single task and across tasks. We compare our approach with the state of the art on an environment with instructions that fully specify the task and an environment in which the instructions are underspecified.

In summary, the contributions of this chapter are:

- the first approach to increase sample efficiency of an RL agent with high-level and underspecified instructions;
- methods for specification, control and learning for options with rich initiation and termination conditions;
- intuitive instruction semantics that allow reuse of options both within a

single task and across multiple tasks;

- state of the art performance in a single-task setting and significant outperformance of the state of the art in zero-shot and multi-task settings across environments with fully specified and underspecified instructions.

After comparing our approach to related work and introducing preliminaries, we introduce our framework in Section 8.4. We detail how instructions are formalized and used for control, then present a learning algorithm in Section 8.5, an experimental evaluation in Section 8.6 and a discussion in Section 8.7.

8.2 Related Work

The literature on improving RL sample efficiency is vast and contains many task- or domain-specific approaches. We limit the discussion here to generic methods for expressing and supplying knowledge to the learner.

8.2.1 Hierarchical RL

Our work uses the expressive formalism of finite state transducers (FSTs) to specify initiation and termination conditions of temporally extended actions and can hence be seen an extension of the options framework [335], see Section 8.3.1. Our framework specifically proposes the use of a, to the best of our knowledge, novel kind of option with non-Markovian initiation and termination conditions, see Section 8.4.3. In the context of hierarchical RL, both sequential [325] and subroutine-based [83] formalisms have been used to define options. Unlike our proposed approach, these formalisms do not allow for interleaving, looping or partial ordering of options.

8.2.2 Classical Planning and RL

High-level control with classical planning and primitive control with RL goes back to Ryan [305] who proposed to use plans obtained from high-level teleo-operators mapping states to suitable behaviors. Another early example used STRIPS planning and was extended with reward shaping [132, 134]. More recently, Yang et al. [394] and Lyu et al. [212] proposed to use an action language from which subtasks are derived. Solutions to these are combined to solve new tasks and are optimized using intrinsic rewards. Illanes et al. [160] introduced the problem of 'taskable RL' and propose a solution based on decomposition. Unfortunately, these works all require a planning goal that specifies the task completely and requires a planning model whereas our approach is robust against underspecified instructions and relies on instructions formalized as an FST which can be specified as e.g. LTL constraints.

8.2.3 Automata, Temporal Logics and RL

The first to recognize that automata can drastically improve RL sample efficiency were Parr and Russell [260]. They proposed a 'hierarchy of abstract machines' to constrain the agent action space. This work was extended by iteratively refining the automata with data [194, 195]. These automata operate on primitive actions and have no abstraction over actions.

Another line of work proposes to specify tasks in temporal logic formulas. These formulas are then converted into a reward function with the aim for the agent is to learn how to satisfy the formula [37, 108, 199, 306], Chapter 7. These works require the full task to be specified whereas we target optimizing an unknown environment reward function using possibly underspecified instructions.

Some works consider decomposition of tasks specified in a temporal logic formula with the option framework. Andreas, Klein and Levine [9] introduced an approach for learning modular behaviors over sequences of subtasks. This approach optimizes an environment reward but does not support looping or interleaving subtasks and requires learning when to switch to a new subtask. Toro Icarte et al. [359] similarly learn a policy per subtask, but infer subtasks from an LTL formula using LTL progression. The same authors propose to learn a policy per state of an automaton representation of the formula [48, 157]. These approaches specify temporally extended behaviors *implicitly*, i.e. there is no transparency at the meta-controller level, whereas we use *explicitly* named options. Reuse of options is therefore limited and their approach may not be applicable to certain zero-shot settings. On top of this, many policies may need to be learned, as the size of the automaton may grow exponentially in the size of the formula. Most importantly, these approaches also require that the entire task is specified upfront, whereas we target optimizing an unknown environment reward with possibly underspecified instructions.

8.3 Preliminaries

8.3.1 Reinforcement Learning

The RL framework can be used to maximize the amount of collected rewards in an environment by selecting an action at each time step [338]. Such problems are formalized as a Markov Decision Problem (MDP) $M : \langle S, A, T, R, \gamma, S_0 \rangle$ with a set of environment states $S = \{s^1, \ldots, s^n\}$, a set of agent actions $A = \{a^1, \ldots, a^m\}$, a probabilistic transition function $T : S \times A \to \mathbb{P}(S)$ function $R : S \times A \times S \to [R_{\min}, R_{\max}]$ with $R_{\min}, R_{\max} \in \mathbb{R}$, a discount factor $\gamma \in [0, 1)$ to balance current and future rewards and S_0 a distribution of initial states at time step $t = 0$. At each time step t, the agent observes an environment state s_t and performs some action $a_t \sim \pi \in \Pi : S \to \mathbb{P}(A)$ and collects reward $r_t = R(s_t, a_t, s_{t+1})$. An optimal policy π^* yields the highest obtainable discounted cumulative rewards. For complex tasks it may be difficult to discover any

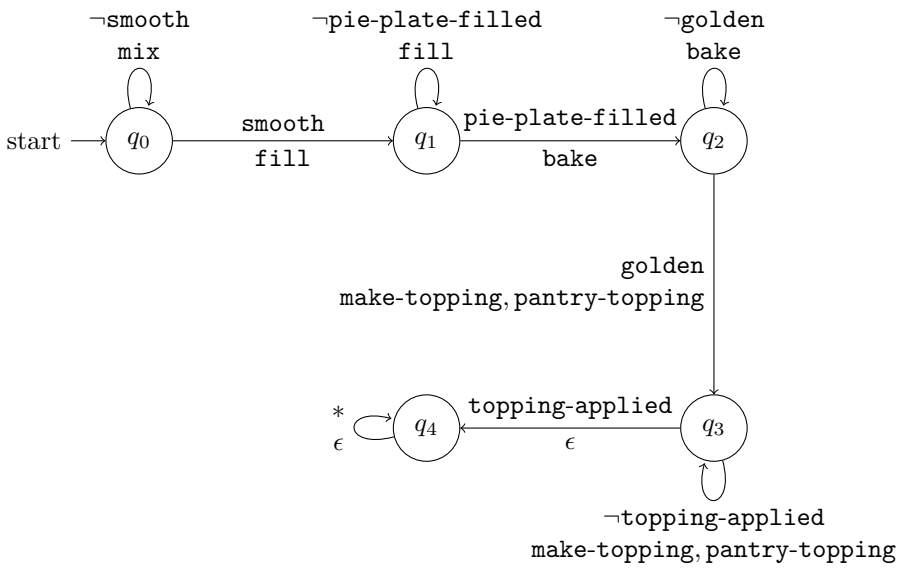


Figure 8.1: Option Machine for the pie recipe from Example 1 with environment events {`smooth`, `pie-plate-filled`, `golden`, `topping-applied`} and options {`mix`, `fill`, `bake`, `make-topping`, `pantry-topping`}.

positive rewards. The agent can be given progressively more complex tasks known as *curriculum learning* [29].

Actor-Critic Methods

Actor-critic (AC) methods optimize a set of weights $\boldsymbol{\theta}$ on which the policy is conditioned: $a \sim \pi(s, \cdot; \boldsymbol{\theta})$ [182, 389]. This *actor* is itself optimized with an estimated state-value $\hat{v}_\pi(s; \mathbf{w})$, conditioned on a second set of weights $\mathbf{w}$ referred to as the *critic*. Both sets of weights can then be optimized with the following update rules for given step sizes $\alpha^{\boldsymbol{\theta}}, \alpha^{\mathbf{w}} > 0$ and a given interaction with the environment (s_t, a_t, r_t, s_{t+1}) and resulting return $g = \sum_{j=t}^{\infty} \gamma^{j-t} R(s_j, a_j, s_{j+1})$ at time t:

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}} \big(\nabla \hat{v}(s_t; \mathbf{w})\big)\big(g - \hat{v}(s_t; \mathbf{w})\big) \tag{8.1}$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha^{\boldsymbol{\theta}} \big(\nabla \log \pi(s_t, a_t; \boldsymbol{\theta})\big) \hat{v}(s_t; \mathbf{w}) \tag{8.2}$$

Options

The option framework introduces an abstraction over the space of actions [335]. The agent selects a 'primitive' action $a \in A$ or 'multi-step' action at each time step. These *options* are formalized as a tuple $\langle \mathcal{I}, \pi, \beta \rangle$ where $\mathcal{I} : S \to \{0, 1\}$ a function indicating in which states the option can be initiated, π a policy that controls the agent when the option is active and $\beta : S \to \{0, 1\}$ a termination function that determines when the option becomes inactive. If the options are

trained with an actor-critic method then each option o can have its own actor $\boldsymbol{\theta}_o$ and critic $\mathbf{w}_o$. We denote the sets of all actors and critics for all options as $\boldsymbol{\Theta}$ and $\mathbf{W}$.

8.3.2 Finite State Transducers

Transducers are a generalization of finite state machines for control and define a mapping between two different types of information. We focus on deterministic FSTs whose output is determined by its current state and input, known in literature as a Mealy machine. We define a FST as a tuple $\varphi : \langle \Sigma, \Omega, Q, I, F, \delta \rangle$ where Σ is a finite input alphabet, Ω a finite output alphabet, Q a finite set of states, $I \subseteq Q$ the set of initial states, $F \subseteq Q$ the set of terminal or final states, $\delta : Q \times (\Sigma \cup \{\epsilon\}) \to Q \times (\Omega \cup \{\epsilon\})$ a transition functions where ϵ the empty string [232]. A FST can be specified in a temporal logic such as LTL and then converted to a FST with out-of-the-box tools [236].

8.4 The Option Machine Framework

In this section we introduce a framework for using underspecified instructions in RL. Specifications for Option Machines (OMs) can be underspecified in two ways. Firstly, the instructions specify what to do at a high level of abstraction rather than at the level of primitive actions. Secondly, a policy following the instructions in OMs is not assumed to always get high environment rewards. This contrasts with most related works, in which following the instructions is equated to high environment rewards. OMs, in contrast, use the environment reward as the canonical definition of the task and leverage instructions for reuse of obtained knowledge, improved exploration and better reward attribution.

Example 1. A recipe gives instructions for a particular type of pie. While each type of pie is a separate task, recipes refer to common steps such as mixing ingredients, pouring, baking etc. Solutions for these steps can be reused across recipes. A recipe may be underspecified and not guarantee a tasty result as baking requires more knowledge than just the recipes.

We now introduce OMs formally from the perspective of a curriculum of tasks. An OM curriculum is defined as a tuple $C : \langle S, A, T, \gamma, \mathcal{R}, P, \Phi, L \rangle$ where S, A, T, γ are defined as usual in RL, see Section 8.3.1. Tasks $\mathcal{R}$ are formalized as a set of environment reward functions, P a probability distribution over tasks $\mathcal{R}$ and instructions Φ as a set of FSTs. Each $\varphi_i \in \Phi$ corresponds to a particular task R_i and has some Σ_i of environment events as its input alphabet. We assume that a function for detecting these events $L : S \to \bigcup_\Phi \Sigma_i$ is available. The main loop can be found in Algorithm 6 and contains components for control and learning.

8.4.1 Instructions as an Option Machine

Our approach uses high-level instructions for a given task. In particular, instructions define traces of high-level behaviors based on high-level descriptions of environment states. This allows for the intuitive formalization of e.g. a recipe.

Example 1. **(cont.)** A recipe 'mix ingredients until smooth, fill pie plate and bake in oven at 180°C until golden. Apply a home-made topping or use a topping from the pantry to finalize the pie.' See Figure 8.1 for an example OM.

High-level descriptions of states consist of events that the agent can detect in the environment. These are formalized a set of atomic propositions AP^I, to which some truth value in $\Sigma : 2^{AP^I}$ can be assigned. Σ corresponds to the input alphabet for the FST associated with the current task. We assume that some function $L : S \to \Sigma$ for detecting these events in states is available, e.g. as a handcrafted or pretrained component. We return to our running example before we look at how events are used for high-level control.

Example 1. **(cont.)** Events {`smooth`, `pie-plate-filled`, `golden`} can be identified from pixel-level states.

High-level Behaviors are actions that take multiple time steps and can be reused across tasks. These are formalized as *options* and denoted with a set of atomic propositions AP^O, to which some truth values in $\Omega : 2^{AP^O}$ can be assigned. At each time step, the permissible options in an OM are determined by this FST output. The current FST state q_t and detected events $L(s_t)$ trigger some FST transition $\delta\big(q_t, L(s_t)\big)$ which produces a new FST state q_{t+1} and an output $\omega_t \in \Omega \cup \epsilon$. The 'true' propositions in ω_t are interpreted as the set of permissible options at that particular time step and are denoted $O_t \subseteq AP^O$. An OM consists of policies associated with options, a FST that specifies which options are permissible and a mechanism to select from these. We discuss selection mechanisms in the next section. If no options are explicitly defined, then this is represented by the empty string $O_t = \{\epsilon\}$. We treat this is a particular output for which the agent uses a dedicated fallback option.

Example 1. **(cont.)** Figure 8.1 shows that `mix` is the only permissible option until the event `smooth` is detected. From this point onward, the option `fill` is permissible until the event `pie-plate-filled` becomes true etc. When the event `golden` has been detected, the two options `make-topping` and `pantry-topping` become permissible simultaneously.

8.4.2 Control with Option Machines

Control in the OM framework assumes a given task R_i with corresponding FST φ_i and has a two-level structure, see Algorithm 7. At the upper, meta-controller level, a suitable option is selected using φ_i. The policy for this option is then executed at the lower level and generates a primitive action $a_t \in A$ to be executed by the agent. In particular, an option is selected based on the FST output. This output defines one or multiple permissible options O_t. For now,

we simply assume these policies to exist and leave the details on how these are optimized from interactions with the environment to Section 8.5.

Example 1. **(cont.)** It may not be clear to a recipe author whether their audience has the right actuators to create a topping. Further, it may not be known whether e.g. pantry toppings are available.

We compare three approaches to select an option from O_t. The first approach assumes a total ordering over all options AP^O which fixes the selected option as the highest-ranked permissible option in O_t. This 'fixed' approach does not incorporate learning in the upper level of control but it comes with the benefit of stability of agent behavior. The other two approaches do incorporate learning in the upper level of decision-making and both use option-specific state-value estimates $\hat{v}(s; \mathbf{w}_o)$. The first of these simply selects an option o from the permissible options O_t greedily:

$$f(O_t, s, \mathbf{W}) = \arg\max_{o \in O_t} \hat{v}(s; \mathbf{w}_o) \tag{8.3}$$

The greedy approach, however, may result in frequent switches between options, e.g. when estimates are inaccurate during early phases of learning or when all permissible options yield a similar return. To mitigate this, we introduce a 'sticky' mechanism that defaults to selecting the previous option o_{t-1} if it is permissible and greedily otherwise:

$$f(O_t, s, o_{t-1}, \mathbf{W}) = \begin{cases} o_{t-1} & \text{if } o_{t-1} \in O_t \\ \arg\max_{o \in O_t} \hat{v}(s; \mathbf{w}_o) & \text{otherwise} \end{cases} \tag{8.4}$$

8.4.3 Reusable Policies and Non-Markovian Options

Policies in the OM framework have names AP^O and can therefore easily be reused within a task or across tasks. For example, the policy for mixing ingredients can be used for mixing both the dough and the filling in a single cake recipe. Additionally, multiple recipes may require mixing dough. Named options enable reuse of policies in e.g. a zero-shot setting where an unseen task can be solved by combining previously encountered options.

The initiation and termination condition of options in our framework are defined by the FST and based on the history of observed events $L(s_0), L(s_1), \ldots, L(s_t)$. These conditions are therefore non-Markovian. This enables powerful yet intuitive control, including looping and interleaving of options.

8.5 Learning with Option Machines

In this section we look at the problem of learning optimal policies for options from environment interactions generated by a sequence of these options. A key

Algorithm 6 Main loop

Input: curriculum $C : \langle S, A, T, \gamma, \mathcal{R}, P, \Phi, L \rangle$, parameterizations $\pi(\cdot; s, \boldsymbol{\theta})$ and $\hat{v}(s; \mathbf{w})$
Parameters: learning steps N, batch size D
Output: set of actors $\boldsymbol{\Theta}$ and set of critics $\mathbf{W}$

$i \leftarrow 0$, $\mathcal{D} \leftarrow \emptyset$, $\boldsymbol{\Theta} \leftarrow \emptyset$, $\mathbf{W} \leftarrow \emptyset$
$\forall o \in AP^O \cup \epsilon$, add random weights $\boldsymbol{\theta}_o$ to $\boldsymbol{\Theta}$, $\mathbf{w}_o$ to $\mathbf{W}$.
while $i < N$ **do**
  while $|\mathcal{D}| < D$ **do**
    sample $(R \in \mathcal{R}, \varphi \in \Phi) \sim P$.
    $d \leftarrow$ rollout for task R and instructions φ. {Alg. 7}
    $\mathcal{D} \leftarrow \mathcal{D} \cup d$.
  end while
  update parameters $\boldsymbol{\Theta}, \mathbf{W}$ with $\mathcal{D}$. {Alg. 8}
  $i \leftarrow i + 1$.
end while
return $\boldsymbol{\Theta}, \mathbf{W}$.

challenge here is to attribute rewards to the appropriate option. If an option was in control at a particular point in time, should future rewards be attributed to this option or not? First, however, we detail how instructions in OMs can be used to guide the agent with shaping rewards.

Shaping rewards are small positive (or negative) intermediate rewards for actions or states that are promising (or to be avoided). These can be defined based on prior knowledge of a good solution. For example, a small positive reward can be given for solving a subtask such as successfully baking a pie crust. The usage of the FST formalism gives a very natural way to delineate subtasks using FST states. In particular, if the FST transitions from some state q to another state $q' \neq q$, a preset shaping reward ρ can be applied to inform the agent that it is progressing according to the instructions. Shaping rewards can be defined naturally in our approach.

We now turn to the problem of attributing rewards to options and propose a method to address it using FST state information. We first consider the simple case where a single option o was active while visiting a FST state q. In this case, a transition from q to another state $q' \neq q$ must have been caused by the actions sampled according to that options' policy $\boldsymbol{\theta}_o$. Hence, future rewards should be used to update that options' policy. The case of multiple options executing before a transition to a new state, however, poses a problem. Reaching the event that triggers this transition requires different policies for the used options. Hence, these interactions should not be used to update both options policies naively. We propose to use all interactions for updating only the policy of the last option executing in a FST state instead.

Example 1. **(cont.)** In Figure 8.1, a single option will execute while visiting

Algorithm 7 Control with an Option Machine

Input: finite-state transducer φ, actors $\boldsymbol{\Theta}$, critics $\mathbf{W}$, labelling $L : S \to \Sigma$
Parameters: shaping reward $\rho \geq 0$
Output: episode d

1: initialize o, $d \leftarrow \emptyset$, $q \leftarrow q_0 \in \varphi$, observe s.
2: **while** q and s are not terminal **do**
3: $(q', O) \leftarrow \delta(q, L(s))$.
4: $o \leftarrow$ select from O. {Equation 8.3 or 8.4}
5: perform action $a \sim \pi(\cdot|s, \boldsymbol{\theta}_o)$.
6: observe r and s'.
7: append (s, o, q, a, r, s') to d.
8: $s \leftarrow s', q \leftarrow q'$.
9: **end while**
10: **return** d.

states $\{q_0, q_1, q_2\}$. During visits to q_3 both `make-topping` and `pantry-topping` may execute (although not at the same time) until the `topping-applied` event is observed. If `pantry-topping` last executes before the event `topping-applied` is observed then this option's policy will be updated during learning.

Algorithm 8 lists a learning algorithm that implements these ideas on reward shaping and reward attribution. First, the final automaton state and active option are extracted and both the discounted cumulative environment return g_e and shaping return g_s are initialized (lines 1-6). In lines 8-12, the last executing option o in a particular FST state q is set as the target option o' to optimize and the shaping rewards are calculated. These are added to the total reward (lines 14-15) and used to update the actor and critic parameters (lines 16-17). The learning algorithm thus leverages FST state information in two ways: firstly, shaping rewards can be supplied to promote exploration and reinforce subtask completion and secondly, interactions are mapped to a single option to ensure that the parameters of the appropriate option are updated.

8.6 Experiments

In this section, we provide an empirical evaluation of OMs in an environment with both fully specified and underspecified instructions. We evaluate OMs in single-task, multi-task and two zero-shot settings to answer the research questions:

1. Do the instructions improve sample efficiency?
2. What are effects of named options and reward shaping?
3. Which option selection method to use?

Algorithm 8 Learning with Option Machines

Input: actors $\boldsymbol{\Theta}$, critics $\mathbf{W}$, episodes $\mathcal{D}$
Parameters: learning rates $\alpha^{\boldsymbol{\theta},\mathbf{w}}$, shaping reward ρ, discount factor γ
Output: updated actors $\boldsymbol{\Theta}$ and critics $\mathbf{W}$

for all $d \in \mathcal{D}$ **do**
  $d' \leftarrow$ reverse episode d.
  $q' \leftarrow q \in d'[0]$.
  $o' \leftarrow o \in d'[0]$. {option to train}
  $g_e \leftarrow 0$. {environment return}
  $g_s \leftarrow 0$. {shaping return}
  for all $(s, o, q, a, r, s') \in d'$ **do**
    if $q \neq q'$ **then**
      $o' \leftarrow o$. {update option to train}
      $g_s \leftarrow \rho$. {add shaping rewards}
    else
      $g_s \leftarrow \gamma g_s$. {discount shaping return}
    end if
    $g_e \leftarrow \gamma g_e + r$. {update environment return}
    $g \leftarrow g_e + g_s$. {total return}
    $\boldsymbol{\theta}_{o'} \overset{+}{\leftarrow} \alpha^{\boldsymbol{\theta}} \big(\nabla \log \pi(a|s, \boldsymbol{\theta}_{o'})\big)\big(g - \hat{v}(s, \mathbf{w}_{o'})\big)$.
    $\mathbf{w}_{o'} \overset{+}{\leftarrow} \alpha^{\mathbf{w}} \big(\nabla \hat{v}(s, \mathbf{w}_{o'})\big)\big(g - \hat{v}(s, \mathbf{w}_{o'})\big)$.
  end for
end for
return $\boldsymbol{\Theta}, \mathbf{W}$

We include versions of OMs for each of the option selection mechanisms described in Section 8.4.2: OM-fixed selects based on an arbitrarily fixed order, OM-greedy selects according to Equation 8.3 and OM-sticky according to Equation 8.4.

8.6.1 Baselines

We compare option machines to three state-of-the-art approaches. Firstly, we include the 'sketch'-based approach proposed by Andreas, Klein and Levine [9]. This approach targets the multi-task setting, uses a *sequence* of subtasks rather than the richer representation proposed here and learns option termination conditions. Secondly, we compare to reward machines (RM) by Icarte et al. [157] which assume that the instructions specify the task fully and require that the training and evaluation subtasks use the same events. This is not the case for the tasks included here and we therefore do not include RM in the zero-shot setting. For all algorithms, we use AC as the base learner and we include a vanilla AC baseline per task in the single-task setting, denoted 'RL'.

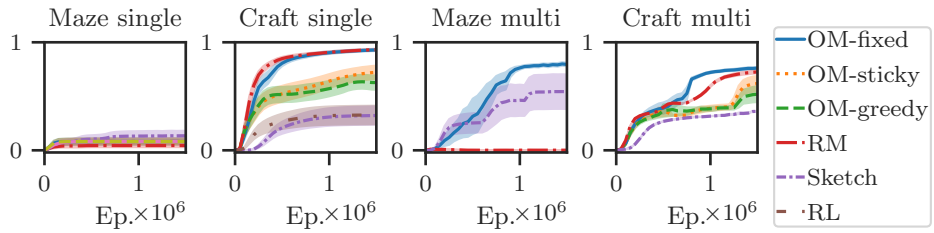


Figure 8.2: Total environment rewards per episode in the single- and multi-task setting on two environments.

			Option Machines		
Env.	Setting	Sketch	Fixed	Greedy	Sticky
maze	isolation	0.09	**0.60**	N/A	N/A
	holdout	0.49	**0.54**		
craft	isolation	0.03	**0.90**	0.74	0.73
	holdout	0.05	**0.86**	0.13	0.22

Table 8.1: Zero-shot total environment reward on 1K test episodes. **Bold** denotes significant best (Mann-Whitney U, $p < 0.01$).

8.6.2 Experimental Setup

Two benchmark environments by [9] are used to evaluate the approach. In the 'craft' environment, items can be obtained by collecting resources such as wood and iron and combining them at workshop locations. Instructions may specify multiple permissible options simultaneously or may fully specify tasks. In the 'maze' environment, the agent must navigate a series of rooms with doors. An event detector describes whether the agent is in a door or not. Critically, it does not differentiate doors leading to the desired room from other doors. As a result, instructions are underspecified. Furthermore, instructions only permit one option at a time. We therefore do not include OM-greedy and OM-sticky in this environment.

An existing curriculum learning setup was used for multi-task learning [9]. Initially, only tasks associated with two options are presented. Once the mean reward on these reaches a threshold of 0.8, this limit is incremented. Tasks within this limit are sampled inversely proportional to the obtained reward. Results were selected with a grid search over hyperparameters. Shaping reward hyperparameters $\rho = 0$ and $\rho = 0.1$ were selected for the maze and craft environment respectively. We report averages over five random seeds. A detailed description of the environments, tasks, hyperparameters etc. can be found in Appendix D.

8.6.3 Results

Single-task Results

The two leftmost graphs in Figure 8.2 show the single-task results on all tasks consisting of more than two options. The maze environment proves too challenging. The reason is its inherent exploration problem which cannot be mitigated by the instructions. Following these does not guarantee solving the task and hence shaping rewards do not help. In the craft environment, shaping is useful: the RM and OM-fixed approaches significantly outperform all others. The usage of named options has negligible effects as RM and OM-fixed perform similarly. Finally, we see a slight advantage of using the sticky option selection over its greedy counterpart.

Multi-task Results

The two rightmost graphs in Figure 8.2 show that the instructions improve sample efficiency as our approach significantly outperforms all baselines. In the maze environment, this can all be attributed to the usage of named options since there are no shaping rewards with $\rho = 0$. Also, note that RMs fail to perform in the multi-task setting because they use the instructions as the full specification of the task. In the craft environment, the instructions *do* fully specify the task and shaping rewards increase sample efficiency. A comparison between OM-fixed and RM indicates that the usage of named options increases sample efficiency significantly. Again, we see that OM-fixed outperforms the other OM variants and that using sticky option selection provides a slight benefit.

Zero-shot Results

We evaluate applicable approaches in two zero-shot settings. In the first setting, policies for all options are trained in isolation and then evaluated on tasks composed of these options. We include all tasks here. In the second setting, policies are trained on a set of training tasks and then evaluated on two unseen, held out, tasks. For OM-based approaches, we execute Algorithm 7 in both settings. Table 8.1 shows that all of the OM versions significantly outperform the baseline in both environments. OM-fixed outperforms all OM versions. The difference here is striking in the holdout case.

The holdout setting is challenging since policies are optimized in the context of tasks other than the evaluation task. As a result, a policy associated with some option o is positively reinforced if it completes a subtask associated with a later option o'. If this subtask is not part of the evaluation task, completing it may harm performance. It could take time and affect later subtasks if these are not commutative. OM-fixed is less susceptible to this failure mode then the other variants, as it uses the same delineation across all episodes. This does not show in the 'isolation' training setting where the greedy and sticky

variants perform significantly better than their counterparts trained in the holdout setting.

8.7 Discussion

We proposed a framework for sample efficient RL with underspecified instructions. These are represented with powerful and intuitive FSTs as a natural way to define shaping rewards and use named options for the reuse of learned behaviors. Experimental evaluations show state of the art performance in a single-task setting and significant outperformance of the state of the art in zero-shot and multi-task settings across environments with fully specified and with underspecified instructions. We have found indications that shaping rewards should not be used when instructions do not cover the task at hand completely but that named options provide a significant benefit. Finally, results indicate that named options significantly increase performance in the multi-task and zero-shot settings.

Future work includes the development of a calculus of instructions for RL with FST operations and the study of ways to derive OMs from interactions to communicate learned strategies with other agents and humans.

Conclusion

9 Conclusion

In this chapter, we conclude this thesis. We revisit the topics that have been addressed and provide answers for the research questions (RQs) posed in Chapter 1. We reflect on the work presented in this thesis and discuss directions for future research. We contributed to five research topics that together shed light on and advance the state of the art of *Reinforcement Learning in human contexts*. We first list the key contributions:

1. a framework for including clinical guideline into RL in Chapter 6. Clinical guideline statements are represented as constraints on the RL state and action spaces and applied via reward shaping or with an action filter. The resulting safe policies only select safe actions and outperform clinicians in a model-based evaluation.

2. a theoretical analysis of RL with high-level symbolic safety constraints and an algorithm that leverages insights from this analysis in Chapter 7. The algorithm significantly outperforms a safe baseline in terms of data efficiency without violating the safety constraints.

3. a framework for RL with instructions, called *Option Machines*, in Chapter 8. Specifically, the instructions specify which long-term behaviors are permissible at a high level of abstraction, making them arguably simple to specify. We proposed algorithms for control and learning within this framework and show that instructions increase data efficiency. Additionally, we show that the framework produces agents that can quickly solve previously unseen tasks which shows that these agents increase controllability.

4. a framework for categorizing personalization problem settings, RL solutions and evaluation strategies in Chapter 2.

5. two RL-based approaches to personalizing dialogue agents, which improve the performance of dialogue agents in Chapter 3.

6. a deep reinforcement learning-based simulation-optimization approach for decision-making in human contexts where the agent and human communicate at the level of rewards and policies in Chapter 5 The approach significantly outperforms a linear programming baseline and the performance difference grows with increasing environment stochasticity.

7. an overview of best practices for collecting user satisfaction ratings for dialogue agents, and a software tool that implements these in Chapter 4.

We continue this chapter with a detailed discussion of these key contributions in the light of the research questions posed in Chapter 1 and treat them in the same order in which these questions were introduced.

9.1 Reinforcement learning for personalization

In Chapters 2 and 3 of Part I and Chapter 6 of Part II, we investigated RL in the particular context of *personalization*. Personalization relates to "the change of the functionality, interface, information access and content or distinctiveness of a system to increase its personal relevance to an individual or a category of individuals" [93]. This allows us to answer research question 1a:

RQ 1a: How has RL been applied to personalization?

The literature survey presented in Chapter 2 shows that RL is an approach to personalization with increasing popularity. It discusses various RL algorithms and describes a framework of personalization problem settings, RL solutions and evaluation strategies. The resulting overview answers how RL can be applied to personalization: first, the particular problem setting is to be considered. We identify eight aspects to characterize problem settings. These aspects of the problem setting can be used to make informed decisions in the design of the solution.

RL solutions were characterized by seven aspects. These aspect together shed light on the nature of the RL solution. Any particular application of RL to personalization will include these aspects to some degree. Our categorization and survey of existing work can be used by researchers and practitioners, to structure their design process, learn from previous endeavors and make use of best practices.

Performance measures of RL solutions to personalization can be obtained using various strategies. Evaluation depends on the problem context and solution design and can be performed in simulation, on real-life data and in a 'live' setting by interacting with users directly. Furthermore, evaluation may include baselines without personalization and non-RL-based personalization strategies in order to attribute effects fairly. We advise to always include these baselines to ensure that the deployment of a RL solution is warranted.

We have used these insights to develop and evaluate a personalized dialogue agent in Chapter 3. In the former, we have implemented multiple versions of

the RL agent based on RL algorithms and the usage of traits of the user in the learning process. In the first usage pattern, traits of the user were included in the agent state representation whereas traits were used to segment the total user population into separate groups in the second pattern. In our evaluation, we included both RL-based approaches, non-RL approaches, approaches with personalization and approaches without personalization. We have found in an empirical evaluation that approaches with RL for personalization outperformed other approaches.

Our insights were additionally used in developing and evaluating an agent for optimizing mechanical ventilation settings in Chapter 6. In this application, aspects of suitability of the outcome (specifically, safety) and upfront knowledge were key aspects (guidelines and prior experiences) in the solution design. We employed a well-known algorithm and extended it with safety guarantees by including the safety constraints in the learning process. We trained and evaluated RL policies on observational data only, an approach known as off-policy policy evaluation that ensures that no additional experimentation on humans is necessary.

9.2 Adaptive dialogue agents

We have made contributions to the field of dialogue agents. This application area is notable as it pertains to a human context in which agents interact with users *directly* and it therefore contains important insights into the usage of RL in human contexts. We proposed two approaches to adaptive dialogue agents based on RL and investigate best practices for collecting dialogue satisfaction scores with third party annotators in Chapters 3 and 4 of Part I, allowing us to answer research question 1b:

> ***RQ 1b: How can we improve and personalize the decision-making in dialogue agents with RL?***

The two novel approaches presented in Chapter 3 show how RL can be successfully applied to personalized dialogue management. With these approaches we extend the state-of-art on decision-making in dialogue agents with personalization. We show that personalization improves the performance of the agents, and conclude that the two approaches each have benefits and drawbacks. Benefits of the first, state space-based approach include the ability to learn to only use context when it is beneficial and does not require segmentation or similarity criteria to be defined upfront; a drawback of this approach is that the inclusion of additional features in the state space representation may make the learning problem more challenging. For example, the learner will have to overcome negative transfer if users with different traits have conflicting desires. The second, segmentation-based approach, does not suffer from this drawback. However, it cannot leverage positive transfer at all because it trains the policies for the segments in isolation. As a result, it may require more data than the state space-based approach.

We compare the approaches in an empirical evaluation with a conversational product recommender by optimizing for an objective measure of quality. Following conventions in the field, we define this measure based on the quality of the provided information and the duration of the dialogue. We rely on a simulator tuned with real-life data. We compare the influence of context and training experiences on performance. In a comparison between the two proposed approaches, we find that performance depends on domain, environment and learning algorithm: no approach dominates the other across settings. However, we do find that learning-based and personalized dialogue managers perform better than or comparable to task-specific approaches as well as a handcrafted gold standard.

While the objective measures used in Chapter 3 is useful for comparing solutions, it is not useful in evaluating the performance of a task-oriented dialogue agent in interactions with real users. Primarily, the objective measure requires that the information need of the user is known at evaluation time while this is not the case in practice. Out of the proposed subjective quality metrics, user satisfaction is typically the ultimate metric to optimize for. User satisfaction ratings are typically obtained from two sources: directly from users and by annotation of third-party raters. Chapter 4 describes how user satisfaction ratings can be obtained and presents a tool for doing so. The tool implements best practices aimed at obtaining high-quality ratings for dialogue: while RL can be used for personalizing adaptive dialogue agents, their performance can be adequately measured with third-party annotators with the proposed tool.

9.3 Operations Management in Human Context

We have investigated the use of deep reinforcement learning (DRL) to address the problem of strategic workforce planning (SWP). SWP provides an interesting problem area from operations management because of (i) its human context and (ii) the indirect nature in which human and agent interact. The interaction is indirect in the sense that interactions are not at the level of states and actions, but rather at the level of goals (formalized as reward) and decision support (formalized as policies). As decisions in SWP directly affect humans, we have to be sure that the problem formalization is well aligned with the intended outcome. Additionally, we need to take the unpredictability of human behavior serious. We have shown that the proposed RL-based solution contributes to both of these goals. Firstly, we have shown that the solution allows the optimization of goals that are more generic and more easy to interpret than the state-of-the-art LP approach. Secondly, we have shown that our proposed solution is more robust to unpredictable human behavior, formalized as stochasticity in the environment. The findings presented in Chapter 5 in Part I help in answering research question 1c.

RQ 1c: How can we improve decision-making with RL

when end-users and agents interact in terms of goals and solutions?

RL can be applied in a simulation-optimization with a suitable simulator using the proposed approach. This framework allows for the use of any black-box simulator that produces a so-called cohort representation of the workforce composition. The proposed approach can optimize objectives composed of arbitrary workforce metrics derived over this cohort representation. Notable examples are metrics that are nonlinear in the workforce compositions such as particular cohorts being within certain bounds or metrics. The proposed approach optimizes such objectives directly, i.e. without requiring any additional transformations such as linearization. This makes the approach easy to use in settings where a human domain expert needs to understand the optimization objective in order to interpret the taken decision.

In the studied use case, we found that the DRL-based approach significantly outperforms a linear programming baseline on such strategic objectives and that the difference grows as stochasticity of the environment increases. In our evaluation, we used historical data of a real-world workforce to derive a simulator. As a result, the SWP problem can be solved with our framework using only (a) a formalization of the workforce goals in terms of metrics over the workforce composition and (b) historical data to derive a simulator from. This enables the use of SWP beyond simple settings in which the target workforce composition is known up-front.

9.4 Safe reinforcement learning

In Part II, we explore the combination of subsymbolic RL and symbolic knowledge. A particularly useful kind of knowledge is what *not* to do in safety-critical human contexts. In particular, we want to provide existing descriptions of (un)safe behavior. Such descriptions may exist at a high level of abstraction, i.e. at an abstraction level above that of atomic environment states and agent actions. We study a setting in which safety constraints are to be avoided both during and after training in Chapters 6 and 7 in order to answer research question 2:

RQ 2: How do safety constraints affect RL learning tasks and how can we improve data efficiency of safe RL?

We have formalized safety constraints from medical guidelines in Chapter 6. In this particular case, the constraints were defined at the level of RL states. We compared two approaches of enforcing the constraints. In the first approach, an unconstrained policy was learned first and then constraints were applied by adjusting the policy. In the second approach, constraints were incorporated in the learning process and specifically in the Q-function definition. We found that the second approach outperformed the former in terms of safety and

expected return based on a model-based evaluation. Additionally, we identified challenges with off-policy policy evaluation in cases where the behavior and evaluation policy are very different – a scenario which may occur from enforcing constraints.
We have then looked into the scenario where safety constraints have a temporal component in Chapter 7. In this scenario, we have formally introduced environments with high-level symbolic safety constraints, analyzed how such constraints impact expected future rewards and showed a relation between expected rewards and the progress toward a goal in a symbolic, automaton representation of the safety constraints. We then proposed an algorithm that leverages this insight. In particular, this algorithm uses symbolic reasoning to infer progress toward a given goal. The learning agent is then encouraged toward progress using potential-based reward shaping. The so-called planning-for-potential algorithm significantly outperformed a safe baseline in terms of data efficiency without violating the safety constraints. The algorithm scaled well as problems became more constrained in an empirical evaluation with real-world constraints and the dialogue agent setting introduced in Chapter 3. We have empirically showed that an RL agent can learn safely and efficiently with the proposed algorithm.

9.5 Reinforcement learning with instructions

A second combination of subsymbolic RL and symbolic knowledge that was studied in Part II entails the usage of symbolic instructions. Similarly to our work on safe RL, these instructions can be formulated at a high level of abstraction, i.e. above the level of atomic states and actions. In contrast to that work, we here focused on instructions that tell the agent on successful behaviors rather than behaviors to avoid. Of note is that these instructions may be incomplete. By supporting such instructions, we alleviate part of the burden of specification. Secondly, this makes the approach applicable to settings where complete instructions are not available. In Chapter 8 we study research question 3:

> ***RQ 3: How can we control RL agents to improve safety and data efficiency?***

We have introduced an approach to RL with instructions called Option Machines (OMs). In the approach, instructions are expressed using the formalism of finite state transducers (FSTs). This makes them easy to specify, reusable across agents and interpretable. The agent then learns behaviors for the steps in the instructions by interacting with the world. These instructions specify which long-term behaviors or *options* are permissible given the history of the interaction. We proposed algorithms for controlling an agent with these instructions and an algorithm for learning behaviors for the options. We train and evaluate the agent in single-task, multi-task and zero-shot settings. We

experimented with the use of shaping rewards and found that these are only useful when the instructions specify the task at hand fully.

Firstly, we found that symbolic instructions increase data efficiency. Secondly, the zero-shot results indicate that the proposed approach produces agents that can quickly be set to complete novel tasks, thus making the agent more controllable. Thirdly, the agent learns long-term behaviors labelled with human-interpretable names given to the agent, such as 'obtain wood' or 'move in the northward direction'. This common language for long-term behaviors can be used for sharing information between the agent and a human, such a human sharing a task description for a previously unseen task composed of known behaviors. This common language can be learned via incomplete instructions, finalizing the contributions for learning with instructions and this thesis in general.

9.6 Discussion & Future Work

We have looked into various applications of RL within human contexts and have proposed several improvements to the field of RL to increase its potential for impact in human contexts. These contributions create avenues for future work which we discuss in this section.

9.6.1 Applications of reinforcement learning in human contexts

In Part I of this thesis, we have looked into various applications of RL in human context. We have surveyed the literature on RL for personalization in Chapter 2 and proposed an application wherein an RL agent directly interacts with users in Chapter 3. In doing so, we followed conventions in the field and used a simulator that was tuned on real-world data for training and evaluation. While this allows full control over experiments and the inclusion of many different solution configurations, it does not conclusively show that the proposed personalization approaches outperform the baselines when trained and evaluated on real-world data. Therefore, we believe a comparison between approaches on real-world data to be an interesting next step. One of the important prerequisites for doing so is that we can guarantee the safety of the agents' behavior, a topic we have contributed to in the second part of this thesis.

Additionally, we have looked at personalizing the agent based on indirect user feedback such as dialogue length and accuracy of recommendation. While the personalization results in better performance according to the objectives of the owner of the agent, these objectives may not align with the users of the agent. Additionally, some users may prefer a more direct approach to personalization of the agent, for example by specifying their objectives such as "short dialogues", "only recommend highly likely items" etc. directly. While this more *direct* approach to personalization is an interesting one, it may be limited to expert users. This is in conflict with the entry level interaction

pattern that dialogue agents offer. Additionally, such direct control measures can be used in conjunction with our proposed indirect approach, for example by making personalization an optional feature and by using a variety of techniques for personalizing different components of the dialogue agent as depicted in Figure 3.1 with different approaches.

Thirdly, we have taken the content of the conversation as the target of personalization to align with the used objective measure of dialogue quality. As our contributions on collecting user satisfaction ratings in Chapter 4 shows, it is also possible to obtain high quality subjective quality metrics with third-party annotators. Recently, an impressive dialogue agent application was created with RL, by fine-tuning a language model for conversations on annotations created by humans [425]. It would have been interesting to combine these works by optimizing dialogue satisfaction scores using a personalized approach. We leave this challenge open for future work.

We have additionally proposed to use RL in a human context with an indirect interaction pattern in Chapter 5 on workforce planning. In this chapter, the user specifies the agents' goal as a reward function composed of multiple components. These different components are linearly combined to form the final optimization objective in our experiments. It would be interesting to treat these components separately by following a multi-objective optimization approach. This would allow the user to learn more about the trade-offs between the different objective components and alleviates the user from having to balance the components up-front. We believe that this extension has become practically feasible with the publication of the code base which includes the learned simulator used in the experiments.

An additional limitation is our choice for a simulation-optimization approach. While suitable SWP strategies are learned with our DRL-based approach, even in challenging scenarios with high stochasticity, their suitability relies on the accuracy of the simulator. If the workforce cannot be simulated accurately, our approach does not apply. We argue in favor of the reliance on a simulator with the following two arguments. Firstly, the use of a separate and black-box simulator is a strength of the approach, as it allows for scenario-based planning: separate scenarios can be modelled in the simulator and the resulting strategies can be analyzed. Secondly, a simulator may be arguably easy to obtain for the following reasons: if knowledge of the workforce dynamics is available, it should be feasible to create a simulator from that knowledge. If such knowledge is not available, we may turn to available historical data describing the workforce. We detail how a simulator can be learned in this case. In the remaining case where both a simulator and historical data are lacking, we argue that little can be done to start with.

Although we have contributed to the application of RL in human context in these important application areas, severe challenges remain. In particular, it is currently poorly understood how to best explain agent decision-making to end users. Here we can identify several separate challenges. Firstly, decisions made by RL agents *now* can be made because the agent believes they produce effects that will be beneficial *later*. Faithfully including this temporal com-

ponent may require lengthy explanations if we communicate about decisions at the level of states and actions. A possible solution here may be the use of temporal abstractions in explanations. Part II contains contributions towards human-understandable temporal abstractions for RL when instructions are available. However, it may be necessary to provide these abstractions also when instructions are not available. An interesting research direction is the study of learning interpretable temporal abstractions over actions. This is a largely unexplored direction of research where the most basic of questions seems so far unanswered: what are desiderata of such abstractions? How to obtain these? Can they be obtained without additional input, such as instructions, and what kind of additional input is necessary?

9.6.2 Subsymbolic RL and symbolic knowledge

Part II explored the combination of subsymbolic RL with symbolic knowledge. This is an exciting and relatively novel field, in which various opportunities for future work exist. Since we cannot include all possibilities, we here restrict ourselves to opportunities that have a reasonably close connection to our own contributions to this field. In relation to the study of safe and efficient RL, we include three major directions for future work. Motivated by our empirical results showing the diverse impact of the various constraints on learner performance, we propose to continue the theoretical analysis on the impact of safety constraints on learning. Theoretical insights may enable us to say, a priori, whether a particular safety constraint will improve or harm the learners' performance and to what degree. It would be particularly interesting to compare semantic and syntactic complexity measures of the safety constraint.

We have studied a setting in which the constraints remain constant over time. However, this is not realistic as new insights on the safety of agent behaviors will have to be incorporated into the agent design. It would therefore also be interesting to study the reuse of past experiences when safety constraints change. Can we simply reuse these experiences in training policies for the new set of constraints? If not, can we 'replay' the behaviors and update the agents' policies using the new constraints? Reuse of data may yield new and safe policies in a more data efficient manner when compared to simply retraining the agent from scratch.

Additionally, it would be interesting to relax some of the assumptions made in Chapter 7. In particular, we believe that the assumption that a fully accurate labelling function is available may not always hold. If these are not available, then either a probabilistic model-checking approach may be required to compute the shield or an alternative approach to shielding is necessary. This will in turn make determining the distance to the goal more complicated as the shield state is now a random variable. However, approaches from RL may be used to estimate an expectation over the distance used in the planning for potential algorithm. Further study is warranted in order to test whether these ideas are fruitful.

We continue our discussion by focusing on the Option Machine (OM) frame-

work proposed in Chapter 8. A key result of this work is that agents can learn to map symbols to temporally abstracted behaviors from instructions. Although the results show that this can be beneficial for data efficiency and that agents can reuse previously learned behaviors in this way, the way in which subtasks are described within the framework is still limited. Another approach would include more rich descriptions of the subtasks based on e.g. common-sense knowledge. For example, it would be interesting to convey the information that if wood (or any other material) is used to make a fire, it can then no longer be used for other purposes. An alternative example is that a pair of scissors can be reused after cutting a piece of paper with it. The agent would be able to use such knowledge to better plan its actions if these invariants were available in a format that allows for automated reasoning. Such a common sense *calculus* of instructions should be usable across different tasks and domains.

An additional interesting avenue for future work that the OM framework unlocks is the study of instructions in a multi-agent setting. Considering that, with the OM framework, agents can learn named options, we can see that it might be possible for agents to give *each other* instructions rather than relying on humans for doing so. There are various interesting ways to implement this general idea, e.g. a setting where the agents learn named options from scratch by communicating about these with each other and a setting in which agents that have different skills collaborate with each other by dividing the subtasks governed by a global OM.

Finally, it is necessary to study the interpretability and controllability of all the approaches proposed in this part in the context of RL end users. This may show how the formal approaches proposed in this part need to be ameliorated or extended with e.g. additional levels of abstraction in order to make useful tools to end users. For example, it may be possibldue to construct a graphical user interface to specify instructions or to construct a structured natural language for specifying safety constraints and agent goals in a user-friendly way. These higher-order specifications can then be compiled into the formalisms used in this thesis such as LTL and FSTs.

Appendices

A

A

A

Appendix A

This is an appendix to Chapter 2. It contains details of the systematic literature review described therein, specifically the queries that were used to generate the initial identification of papers in Figure 2.3 and a tabular view of the data after qualitative synthesis in the same figure.

Queries

All queries as run on June 6, 2018 on the databases included in the review.

Listing A.1: Query for Scopus Database

```
TITLE-ABS-KEY(
("reinforcement learning" OR "contextual bandit") AND
("personalization" OR "personalized" OR "personal" OR
"personalisation" OR "personalised" OR
"customization" OR "customized" OR
"customised" OR "customised" OR
"individualized" OR "individualised" OR "tailored"))
```

Listing A.2: Query for IEEE Xplore Database Command Search

```
(((reinforcement learning) OR contextual bandit) AND
(personalization OR personalized OR personal OR
personalisation OR personalised OR
customization OR customized OR customised OR customised OR
individualized OR individualised OR tailored))
```

Listing A.3: Query for ACM DL Database

```
("reinforcement learning" OR "contextual bandit") AND
(personalization OR personalized OR personal OR
```

```
personalisation OR personalised OR
customization OR customized OR customised OR customised OR
individualized OR individualised OR tailored)
```

Listing A.4: First Query for DBLP Database

```
reinforcement learning
(personalization|personalized|personal|
personalisation|personalised|
customization|customized|customised|customised|
individualized|individualised|tailored)
```

Listing A.5: Second Query for DBLP Database

```
contextual bandit
(personalization|personalized|personal|
personalisation|personalised|
customization|customized|customised|customised|
individualized|individualised|tailored)
```

Listing A.6: First Query for Google Scholar Database

```
allintitle: "reinforcement learning"
personalization OR personalized OR personal OR
personalisation OR personalised OR
customization OR customized OR
customised OR customised OR
individualized OR individualised OR tailored
```

Listing A.7: Second Query for Google Scholar Database

```
allintitle: "contextual bandit"
personalization OR personalized OR personal OR
personalisation OR personalised OR
customization OR customized OR
customised OR customised OR
individualized OR individualised OR tailored
```

A.1 Tabular view of data

Table A.1: Table containing all included publications. The first column refers to the data items in Table 2.2.

#	Value	Publications
1	n	[1, 8, 18, 21, 24, 32, 35, 36, 38, 42, 49, 51–54, 56, 58, 61, 62, 64, 68–71, 76, 77, 79, 82, 89, 96, 112, 117, 119, 122, 129, 152, 155, 156, 162, 166, 173, 174, 185, 186, 193, 196, 198, 200, 202, 207, 209, 210, 214, 218–221, 223, 224, 226–228, 233–235, 242, 244, 245, 249, 250, 252, 256, 257, 259, 262, 273, 278, 280, 282–284, 290, 296, 302, 307, 314, 315, 317, 319, 329, 330, 333, 334, 342–346, 348, 349, 352, 353, 360–362, 368, 378, 379, 382, 390, 393, 395–397, 403, 407, 409, 411, 413–421]
	y	[4, 14, 16, 59, 60, 87, 97, 99–101, 103, 107, 125, 127, 146, 153, 188, 201, 208, 216, 253, 268–270, 287, 297, 318, 327, 331, 363–365, 373, 377, 380, 402, 404, 405, 412]
2	n	[4, 14, 18, 24, 42, 53, 62, 64, 68, 77, 82, 87, 97, 103, 125, 127, 188, 201, 208, 224, 228, 234, 253, 259, 262, 268, 283, 290, 331, 373, 377, 380, 402–404, 412]
	y	[1, 8, 16, 21, 32, 35, 36, 38, 49, 51, 52, 54, 56, 58–61, 69–71, 76, 79, 89, 96, 99–101, 107, 112, 117, 119, 122, 129, 146, 152, 153, 155, 156, 162, 166, 173, 174, 185, 186, 193, 196, 198, 200, 202, 207, 209, 210, 214, 216, 218–221, 223, 226, 227, 233, 235, 242, 244, 245, 249, 250, 252, 256, 257, 269, 270, 273, 278, 280, 282, 284, 287, 296, 297, 302, 307, 314, 315, 317–319, 327, 329, 330, 333, 334, 342–346, 348, 349, 352, 353, 360–365, 368, 378, 379, 382, 390, 393, 395–397, 405, 407, 409, 411, 413–421]
3	n	[1, 4, 8, 14, 16, 18, 21, 24, 32, 35, 36, 49, 51–54, 56, 59–62, 64, 68, 69, 79, 82, 87, 89, 96, 97, 99–101, 103, 112, 119, 122, 125, 127, 129, 152, 153, 156, 162, 166, 173, 174, 185, 188, 193, 198, 200–202, 208, 209, 214, 218–221, 223, 226–228, 233, 234, 242, 244, 245, 249, 252, 253, 256, 257, 259, 268–270, 273, 280, 282–284, 287, 290, 296, 307, 314, 315, 317–319, 327, 329–331, 333, 334, 342–346, 348, 349, 353, 360, 362, 368, 377–380, 382, 390, 393, 395–397, 402–405, 407, 409, 412–421]
	y	[38, 42, 58, 70, 71, 76, 77, 107, 117, 146, 155, 186, 196, 207, 210, 216, 224, 235, 250, 262, 278, 297, 302, 352, 361, 363–365, 373, 411]

#	Value	Publications
4	n	[4, 8, 14, 16, 18, 21, 24, 32, 35, 36, 38, 42, 49, 51–54, 56, 58–62, 64, 68–71, 76, 77, 79, 82, 87, 89, 96, 97, 99–101, 103, 107, 112, 117, 119, 122, 125, 129, 146, 152, 153, 155, 156, 166, 173, 174, 185, 186, 188, 193, 196, 200–202, 207–210, 214, 216, 218–221, 223, 224, 226–228, 233–235, 242, 244, 245, 249, 250, 252, 253, 256, 257, 259, 262, 268–270, 273, 278, 280, 282–284, 287, 290, 296, 297, 302, 314, 315, 317–319, 327, 329, 331, 333, 334, 342–346, 348, 349, 352, 353, 360–363, 365, 368, 373, 377–379, 382, 390, 393, 395–397, 402–405, 407, 409, 411–418, 420, 421]
	y	[1, 127, 162, 198, 307, 330, 364, 380, 419]
5	n	[1, 4, 14, 18, 21, 24, 36, 42, 49, 51–54, 56, 59–61, 64, 68, 69, 82, 87, 89, 96, 97, 107, 112, 122, 125, 129, 146, 152, 155, 156, 162, 166, 173, 174, 185, 188, 193, 196, 198, 200–202, 207–210, 214, 216, 219, 220, 226–228, 234, 235, 242, 244, 245, 249, 253, 256, 257, 259, 262, 269, 270, 273, 278, 282–284, 287, 290, 296, 297, 302, 307, 314, 315, 317, 319, 327, 329–331, 333, 342, 343, 345, 346, 348, 352, 353, 360–362, 364, 365, 373, 377–380, 382, 390, 393, 395, 403, 404, 407, 409, 412–421]
	y	[8, 16, 32, 35, 38, 58, 62, 70, 71, 76, 77, 79, 99–101, 103, 117, 119, 127, 153, 186, 218, 221, 223, 224, 233, 250, 252, 268, 280, 318, 334, 344, 349, 363, 368, 396, 397, 402, 405, 411]
6	n	[1, 4, 8, 16, 18, 24, 32, 35, 42, 51–53, 56, 58, 59, 64, 68, 69, 71, 76, 77, 79, 82, 87, 89, 96, 97, 103, 107, 117, 119, 122, 146, 152, 153, 155, 156, 166, 173, 174, 185, 186, 193, 196, 201, 210, 214, 216, 218–221, 223, 224, 227, 233–235, 245, 249, 250, 252, 256, 257, 259, 268, 270, 278, 280, 283, 290, 296, 302, 307, 317–319, 329, 331, 333, 334, 343, 360–363, 365, 368, 377, 382, 390, 393, 407, 411, 413–417, 421]
	y	[14, 21, 36, 38, 49, 54, 60–62, 70, 99–101, 112, 125, 127, 129, 162, 188, 198, 200, 202, 207–209, 226, 228, 242, 244, 253, 262, 269, 273, 282, 284, 287, 297, 314, 315, 327, 330, 342, 344–346, 348, 349, 352, 353, 364, 373, 378–380, 395–397, 402–405, 409, 412, 418–420]
7	n	[1, 21, 24, 32, 52–54, 56, 58, 61, 62, 64, 68, 70, 79, 82, 87, 89, 97, 112, 117, 122, 129, 146, 174, 186, 188, 193, 196, 201, 207, 210, 219, 220, 223, 224, 234, 235, 244, 249, 252, 253, 259, 268, 273, 278, 280, 283, 290, 297, 307, 314, 319, 327, 329, 331, 360–362, 364, 365, 373, 382, 404, 407, 411, 414–416]

#	Value	Publications
	y	[4, 8, 14, 16, 18, 35, 36, 38, 42, 49, 51, 59, 60, 69, 71, 76, 77, 96, 99–101, 103, 107, 119, 125, 127, 152, 153, 155, 156, 162, 166, 173, 185, 198, 200, 202, 208, 209, 214, 216, 218, 221, 226–228, 233, 242, 245, 250, 256, 257, 262, 269, 270, 282, 284, 287, 296, 302, 315, 317, 318, 330, 333, 334, 342–346, 348, 349, 352, 353, 363, 368, 377–380, 390, 393, 395–397, 402, 403, 405, 409, 412, 413, 417–421]
8	n	[38, 49, 53, 56, 60, 62, 96, 107, 117, 125, 162, 198, 208, 209, 216, 219, 242, 256, 262, 283, 296, 342, 345, 353, 365, 378–380, 382, 393, 405, 409, 417, 419]
	y	[1, 4, 8, 14, 16, 18, 21, 24, 32, 35, 36, 42, 51, 52, 54, 58, 59, 61, 64, 68–71, 76, 77, 79, 82, 87, 89, 97, 99–101, 103, 112, 119, 122, 127, 129, 146, 152, 153, 155, 156, 166, 173, 174, 185, 186, 188, 193, 196, 200–202, 207, 210, 214, 218, 220, 221, 223, 224, 226–228, 233–235, 244, 245, 249, 250, 252, 253, 257, 259, 268–270, 273, 278, 280, 282, 284, 287, 290, 297, 302, 307, 314, 315, 317–319, 327, 329–331, 333, 334, 343, 344, 346, 348, 349, 352, 360–364, 368, 373, 377, 390, 395–397, 402–404, 407, 411–416, 418, 420, 421]
10	1	[1, 8, 18, 21, 35, 51–54, 56, 60, 61, 64, 68, 70, 71, 77, 79, 89, 96, 112, 117, 122, 152, 153, 155, 156, 162, 173, 174, 186, 188, 193, 198, 201, 202, 207, 208, 219, 223, 224, 226, 235, 244, 250, 252, 256, 259, 280, 282–284, 302, 307, 314, 318, 319, 329, 331, 333, 334, 342–346, 348, 352, 353, 361–364, 368, 373, 377–380, 395–397, 402, 404, 407, 411, 414, 415, 417, 420, 421]
	1/group	[32, 69, 200, 209, 214, 220, 221, 273, 382, 390, 416]
	1/person	[14, 16, 24, 36, 38, 42, 49, 58, 59, 62, 76, 87, 97, 99–101, 103, 107, 119, 125, 127, 129, 146, 166, 185, 196, 210, 216, 218, 227, 228, 233, 234, 245, 249, 253, 257, 262, 268–270, 278, 287, 290, 296, 297, 315, 317, 327, 349, 360, 365, 393, 403, 405, 409, 412, 413, 418]
	multiple	[4, 82, 242, 330, 419]
11	not used	[14, 16, 18, 24, 32, 35, 42, 51, 53, 56, 59–61, 64, 68–70, 76, 77, 89, 97, 103, 112, 119, 122, 125, 152, 153, 155, 162, 173, 174, 185, 188, 193, 201, 208, 209, 216, 219, 226, 227, 235, 244, 245, 256, 257, 259, 269, 283, 297, 302, 314, 319, 329, 331, 334, 342, 343, 349, 352, 360, 368, 373, 377, 380, 393, 395, 402, 404, 413, 418, 420]
	other	[4, 36, 49, 127, 129, 233, 234, 242, 268, 278, 282, 287, 345, 348, 407, 411, 419]

A

#	Value	Publications
	state representation	[1, 8, 21, 38, 52, 54, 58, 62, 71, 79, 82, 87, 96, 99–101, 107, 117, 146, 156, 166, 186, 196, 198, 200, 202, 207, 210, 214, 218, 220, 221, 223, 224, 228, 249, 250, 252, 253, 262, 270, 273, 280, 284, 290, 296, 307, 315, 317, 318, 327, 330, 333, 344, 346, 353, 361–365, 378, 379, 382, 390, 396, 397, 403, 405, 409, 412, 414–417, 421]
12	batch	[1, 60, 61, 68, 82, 87, 96, 99–101, 103, 127, 162, 186, 188, 200, 210, 219–221, 223, 224, 228, 250, 252, 268, 280, 282, 290, 327, 330, 333, 334, 342, 343, 346, 348, 349, 352, 353, 361, 378, 379, 395, 405, 407, 414–417, 419, 421]
	n	[393]
	online	[14, 16, 35, 36, 38, 42, 49, 51, 54, 58, 59, 62, 71, 77, 79, 97, 107, 117, 119, 122, 125, 146, 156, 166, 173, 174, 185, 196, 198, 201, 202, 207–209, 216, 218, 227, 233–235, 244, 245, 253, 256, 257, 262, 269, 270, 287, 296, 297, 302, 307, 314, 315, 317, 318, 329, 344, 345, 363, 365, 368, 373, 380, 396, 397, 402–404, 412, 413, 418, 420]
	other	[76, 129, 152, 242, 249, 273, 382, 409]
	unknown	[4, 8, 18, 21, 24, 32, 52, 53, 56, 64, 69, 70, 89, 112, 153, 155, 193, 214, 226, 259, 278, 283, 284, 319, 331, 360, 362, 364, 377, 390, 411]
13	n	[1, 4, 14, 21, 36, 51–54, 56, 59–62, 68, 70, 82, 89, 112, 129, 162, 173, 174, 186, 188, 193, 198, 200–202, 207, 208, 214, 219, 220, 223, 224, 226–228, 235, 244, 250, 252, 253, 256, 257, 259, 269, 273, 280, 282, 284, 287, 290, 307, 314, 315, 317, 319, 327, 329, 330, 346, 348, 349, 352, 353, 362, 364, 373, 377, 378, 390, 393, 395–397, 402, 404, 407, 409, 411, 412, 414, 417, 418, 420]
	y	[8, 16, 18, 24, 32, 35, 38, 42, 49, 58, 64, 69, 71, 76, 77, 79, 87, 96, 97, 99–101, 103, 107, 117, 119, 122, 125, 127, 146, 152, 153, 155, 156, 166, 185, 196, 209, 210, 216, 218, 221, 233, 234, 242, 245, 249, 262, 268, 270, 278, 283, 296, 297, 302, 318, 331, 333, 334, 342–345, 360, 361, 363, 365, 368, 379, 380, 382, 403, 405, 413, 415, 416, 419, 421]
14	n	[4, 8, 14, 16, 18, 24, 32, 35, 38, 42, 49, 51–54, 56, 58–60, 62, 64, 69, 71, 76, 77, 96, 97, 99–101, 103, 107, 112, 117, 119, 122, 125, 127, 129, 153, 155, 156, 166, 173, 174, 185, 193, 196, 202, 209, 210, 214, 216, 218, 221, 233–235, 245, 249, 256, 259, 262, 268–270, 278, 283, 287, 296, 297, 302, 307, 314, 315, 318, 329, 331, 342, 360, 362–365, 368, 373, 377, 379, 382, 393, 402–405, 413, 415–418, 420]

#	Value	Publications
	y	[1, 21, 36, 61, 68, 70, 79, 82, 87, 89, 146, 152, 162, 186, 188, 198, 200, 201, 207, 208, 219, 220, 223, 224, 226–228, 242, 244, 250, 252, 253, 257, 273, 280, 282, 284, 290, 317, 319, 327, 330, 333, 334, 343–346, 348, 349, 352, 353, 361, 378, 380, 390, 395–397, 407, 409, 411, 412, 414, 419, 421]
15	n	[1, 4, 16, 18, 21, 24, 32, 35, 36, 42, 49, 52, 53, 56, 58, 60, 61, 64, 68–71, 76, 77, 79, 82, 87, 89, 96, 97, 103, 107, 117, 119, 122, 127, 146, 153, 155, 162, 166, 185, 186, 188, 198, 200, 201, 207–210, 214, 218–221, 223, 224, 226–228, 233, 234, 242, 244, 245, 249, 250, 252, 253, 257, 259, 262, 268, 270, 273, 278, 280, 282, 283, 290, 296, 297, 302, 317–319, 327, 330, 333, 334, 342–345, 348, 352, 353, 360–365, 368, 377–379, 382, 390, 393, 395–397, 405, 407, 411, 412, 414–417, 419, 421]
	y	[8, 14, 38, 51, 54, 59, 62, 99–101, 112, 125, 129, 152, 156, 173, 174, 193, 196, 202, 216, 235, 256, 269, 284, 287, 307, 314, 315, 329, 331, 346, 349, 373, 380, 402–404, 409, 413, 418, 420]
16	n	[1, 4, 14, 21, 36, 51–54, 56, 59–62, 68, 82, 89, 112, 129, 146, 156, 162, 173, 174, 186, 188, 193, 198, 200, 202, 207, 208, 214, 219, 220, 223, 224, 226–228, 235, 244, 250, 252, 253, 256, 257, 259, 269, 273, 280, 282, 284, 287, 290, 307, 314, 315, 317, 319, 327, 329, 330, 346, 348, 349, 352, 353, 362, 364, 365, 373, 377, 378, 390, 393, 395–397, 402, 404, 409, 411, 412, 414, 417, 418, 420]
	y	[8, 16, 18, 24, 32, 35, 38, 42, 49, 58, 64, 69–71, 76, 77, 79, 87, 96, 97, 99–101, 103, 107, 117, 119, 122, 125, 127, 152, 153, 155, 166, 185, 196, 201, 209, 210, 216, 218, 221, 233, 234, 242, 245, 249, 262, 268, 270, 278, 283, 296, 297, 302, 318, 331, 333, 334, 342–345, 360, 361, 363, 368, 379, 380, 382, 403, 405, 407, 413, 415, 416, 419, 421]
17	n	[4, 8, 14, 16, 18, 24, 32, 35, 38, 42, 49, 51–54, 56, 58–62, 64, 69–71, 76, 77, 96, 97, 99–101, 103, 107, 112, 117, 119, 122, 125, 127, 129, 153, 155, 156, 166, 173, 174, 185, 193, 196, 201, 202, 209, 210, 214, 216, 218, 221, 233–235, 244, 245, 249, 256, 259, 262, 268–270, 278, 283, 287, 296, 297, 302, 307, 314, 315, 318, 329–331, 333, 342, 349, 360, 362–365, 368, 373, 377, 379, 382, 393, 402–405, 407, 413, 415, 417, 418, 420]
	y	[1, 21, 36, 68, 79, 82, 87, 89, 146, 152, 162, 186, 188, 198, 200, 207, 208, 219, 220, 223, 224, 226–228, 242, 250, 252, 253, 257, 273, 280, 282, 284, 290, 317, 319, 327, 334, 343–346, 348, 352, 353, 361, 378, 380, 390, 395–397, 409, 411, 412, 414, 416, 419, 421]

#	Value	Publications
18	n	[1, 4, 16, 18, 21, 24, 32, 35, 36, 42, 49, 52, 53, 56, 58, 64, 68–71, 76, 77, 79, 82, 87, 89, 96, 97, 103, 107, 117, 119, 122, 127, 146, 155, 162, 166, 185, 186, 188, 198, 200, 207–210, 214, 218–221, 223, 224, 226–228, 233, 234, 242, 245, 249, 250, 252, 253, 257, 259, 262, 268, 270, 273, 278, 280, 282, 283, 290, 296, 302, 317–319, 327, 333, 342–345, 348, 352, 353, 360–365, 368, 377–379, 382, 390, 393, 395, 405, 407, 411, 412, 414–416, 421]
	y	[8, 14, 38, 51, 54, 59–62, 99–101, 112, 125, 129, 152, 153, 156, 173, 174, 193, 196, 201, 202, 216, 235, 244, 256, 269, 284, 287, 297, 307, 314, 315, 329–331, 334, 346, 349, 373, 380, 396, 397, 402–404, 409, 413, 417–420]
19	n	[1, 4, 16, 35, 36, 42, 49, 51–54, 56, 61, 62, 68–71, 77, 87, 89, 96, 97, 99–101, 107, 117, 127, 146, 152, 155, 156, 162, 166, 173, 174, 188, 193, 200, 202, 207–210, 214, 216, 218–220, 226–228, 233–235, 244, 245, 252, 253, 256, 259, 262, 268, 270, 273, 278, 282, 287, 290, 296, 297, 302, 307, 314, 318, 319, 329, 333, 334, 342–344, 352, 353, 360, 362–365, 368, 377, 379, 382, 390, 393, 395, 402, 404, 405, 413–415, 417–419, 421]
	y	[8, 14, 18, 21, 24, 32, 38, 58–60, 64, 76, 79, 82, 103, 112, 119, 122, 125, 129, 153, 185, 186, 196, 198, 201, 221, 223, 224, 242, 249, 250, 257, 269, 280, 283, 284, 315, 317, 327, 330, 331, 345, 346, 348, 349, 361, 373, 378, 380, 396, 397, 403, 407, 409, 411, 412, 416, 420]
20	n	[1, 4, 8, 14, 16, 18, 21, 24, 35, 36, 42, 49, 51–54, 56, 58–60, 62, 64, 68–71, 76, 77, 82, 87, 89, 96, 97, 99–101, 103, 107, 112, 119, 122, 125, 127, 129, 146, 152, 155, 156, 162, 166, 173, 186, 193, 198, 200, 202, 207, 210, 214, 216, 218–220, 223, 224, 226, 227, 234, 235, 244, 245, 249, 250, 256, 259, 262, 269, 273, 278, 283, 287, 290, 296, 297, 302, 318, 319, 329–331, 334, 344, 345, 348, 349, 352, 353, 360, 362–365, 373, 377–379, 382, 390, 393, 395, 402, 404, 405, 407, 413, 415, 416, 418–420]
	y	[32, 38, 61, 79, 117, 153, 174, 185, 188, 196, 201, 208, 209, 221, 228, 233, 242, 252, 253, 257, 268, 270, 280, 282, 284, 307, 314, 315, 317, 327, 333, 342, 343, 346, 361, 368, 380, 396, 397, 403, 409, 411, 412, 414, 417, 421]
Domain	Commerce	[1, 36, 52, 82, 96, 152, 156, 200, 208, 214, 216, 218, 228, 242, 256, 284, 343, 346, 348, 352, 353, 382, 402, 403, 407, 412, 414, 417]
	Communication	[70, 185, 193, 307]

#	Value	Publications
	Domain Independent	[38, 49, 62, 224, 245, 249, 262, 344, 396, 397, 405]
	Education	[54, 56, 61, 89, 112, 129, 146, 153, 173, 174, 188, 209, 219–221, 259, 287, 290, 318, 319, 333, 334, 378, 379, 395]
	Energy	[166, 233, 234, 268, 368, 413]
	Entertainment	[8, 18, 35, 60, 64, 99–101, 103, 122, 125, 162, 198, 201, 226, 227, 244, 269, 282, 283, 296, 317, 330, 331, 342, 345, 380, 404, 409, 419]
	Health	[4, 14, 16, 21, 53, 68, 69, 71, 76, 77, 79, 87, 117, 119, 127, 186, 196, 207, 210, 223, 235, 250, 252, 253, 273, 278, 280, 297, 302, 314, 327, 360–364, 373, 393, 411, 415, 416, 418, 420, 421]
	Other	[42, 51, 315, 365, 377]
	Smart Home	[59, 97, 202, 349]
	Transport	[24, 32, 58, 107, 155, 257, 270, 329, 390]

A

Appendix B

This appendix details the implementation and configuration for all experiments described in Chapter 5.

B.1 Model Details and Training Setup

We used the Proximal Policy Optimiziation algorithm by Schulman et al. [313] with the hyperparameters detailed in Table B.1. We considered two architectures: one with shared network weights and another with completely detached network weights. We considered between one to three NN layers with between $[16, 256]$ nodes each and chose the an architecture with a separate actor of two layers of 128 nodes and separate critic with two layers of 256 NN nodes. These hyperparameters were chosen based on results of a grid search.

The agent was trained as followed. First, a maximum number of training steps was specified. In each episode, a random start state in the neighborhood of the initial headcount of the organization X_0 was generated by uniformly sampling a state close to X_0. The episode then runs for a fixed number of time steps, after which the environment resets to a new random start state. This process repeats until the total number of training samples has been reached. After a fixed number of time steps, the agent is evaluated on an evaluation environment. This evaluation environment is governed by the same dynamics as the one used for training, but always starts at the same state X_0. When the training process has terminated, we test the trained model on the evaluation environment with a fixed starting state X_0 and to determine the quality of the final model.

Param.	Description	Considered	Final
α	Adam learning rate	$[0.001, 0.00001]$	0.0003
T	Episode length	$[8, 60]$	16
N_{steps}	Rollout buffer size	$[8, 2048]$	256
B	Batch size	$[4, 64]$	32
K	Number of epochs	$[2, 8]$	4
γ	Discount factor	$[0.90, 0.9990]$	0.90
λ	GAE factor	$[0.90, 0.990]$	0.90
ϵ	PPO clipping range	$[0.1, 0.3]$	0.25
c_2	Entropy coefficient	$\{0.0, 0.01\}$	0.01
c_1	Value function coefficient	$[0.0, 1.0]$	1.0

Table B.1: PPO hyperparameter selection for all SWP experiments.

Appendix C

This appendix supports Chapter 6. It first details the data, preprocessing, and modeling of the action space and then describes outcomes on tests of significance for the results.

C.1 Cohort and Pre-processing details

Number of ICUs	5
Acquisition timespan	2001-2012
Number of included patients	7659
Number of included ventilation events	8799
Age	65.67 (53.19-76.44) years
Body weight	86.24 ± 24.89 kg
Ideal body weight	63.38 ± 12.93 kg
Sex, female	3813 (43.33%)
Sex, male	4986 (56.67%)
90-day mortality	34.50%
in-hospital mortality	25.73%
LOS ICU	7.58 (4.29-13.58) days
LOS hospital	14.62 (8.58-13.58)days
PEEP	6.4 ± 2.18 cmH_2O
FiO_2	45.58 ± 7.41%
Vt_{set}	8.40 ±7.19 mL/kg IBW
SOFA at ICU admission	4.20 ±3.32 points

Table C.1: Demographic and clinical data of the included patient cohort extracted from MIMIC-III where ranges indicate median and (first quantile, third quantile) and other values are mean ± standard deviation.

	Variable	Window (h)	% Missing		
			Initial	1^{st} Step	2^{nd} Step
demographic	Age	–	0.0	0.0	0.0
	IBW		16.8	16.8	
	Height		16.8	16.8	
	Weight		14.4	14.4	
	ICU readmission		0.0	0.0	
	Elixhauser-vanWalraven		77.2	77.2	
vital signs	SOFA	24	0.0	0.0	0.0
	SIRS	24	0.0	0.0	
	GCS	∗	19.0	1.5	
	HR	∗	1.4	0.6	
	SysBP	∗	2.5	0.6	
	MeanBP	∗	1.9	0.6	
	DiasBP	∗	2.5	0.6	
	ShockIndex	∗	3.2	0.7	
	RR	∗	1.8	0.6	
	SpO_2	∗	2.2	0.6	
	TempC	∗	9.4	1.2	
action	PEEP	8	33.7	25.3	0.0
	FiO_2		26.1	16.6	
	Vt_{set}		33.8	25.4	
other	IV	8	14.1	5.7	0.0
	Urine output	8	14.5	11.6	
	Fluid Balance	8	3.6	2.4	
	Plateau Pressure	8	80.1	7.8	
	Vasopressors (dosage)	24	88.4	74.8	
	PaO_2/FiO_2 ratio	∗	98.3	55.7	

Table C.2: Overview of variables and missing data before and after the first, sample-and-hold imputation step and after the second, k-nearest neighbour imputation step where – denotes that no sample-and-hold imputation was applied for a variable and ∗ denotes that sample-and-hold was applied until the next measurement or the end of the trajectory.

	Name	Window	% Missing		
		(h)	Initial	1^{st} Step	2^{nd} Step
lab measurements	Potassium	∗	95.4	4.2	
	Sodium	∗	95.5	3.9	
	Chloride	∗	95.5	3.4	
	Glucose	∗	95.7	4.8	
	BUN	∗	95.5	2.6	
	Creatinine	∗	95.5	2.6	
	Magnesium	∗	95.5	7.0	
	Calcium	∗	95.9	11.3	
	Ionized Calcium	8	96.4	56.8	
	Calculated Carbon Dioxide†	∗	83.2	9.7	
	Bilirubin	∗	94.0	42.4	
	Albumin	∗	98.4	51.4	
	Hemoglobin	∗	98.9	3.0	0.0
	WBC	∗	95.8	2.9	
	Platelet	∗	95.6	2.5	
	PTT	∗	96.3	8.5	
	PT	∗	96.2	8.0	
	INR	∗	96.2	8.0	
	PH	∗	93.9	8.9	
	PaO_2	∗	97.8	45.8	
	$PaCO_2$†	∗	94.4	12.7	
	Base Excess	∗	94.5	13.0	
	Bicarbonate	∗	95.6	3.4	
	Lactate	∗	96.3	21.2	

Table C.3: Overview of variables and missing data before and after the first, sample-and-hold imputation step and after the second, k-nearest neighbour imputation step where where ∗ denotes that sample-and-hold was applied until the next measurement or the end of the trajectory. †Calculated Carbon (LOINC 34728-6) refers to the total calculated Carbion Dioxide (moles/volume) in the blood whereas $PaCO_2$ (LOINC 11557-6) refers to the measured CO_2 (partial pressure) in blood. These where both included in the approach by Peine et al. [266] and were included in this study for enable easy comparisons.

Variable	#	Range	Variable	#	Range	Variable	#	Range
	1	$[0, 2.5)$		1	$[0, 5)$		1	$[20, 30)$
	2	$[2.5, 5)$		2	$[5, 7)$		2	$[30, 35)$
	3	$[5, 7.5)$		3	$[7, 9)$		3	$[35, 40)$
Vt_{set}	4	$[7.5, 10)$	PEEP	4	$[9, 11)$	FiO_2	4	$[40, 45)$
	5	$[10, 12.5)$		5	$[11, 13)$		5	$[45, 50)$
	6	$[12.5, 15)$		6	$[13, 15)$		6	$[50, 55)$
	7	$[15, \infty)$		7	$[15, \infty)$		7	$[55, \infty)$

Table C.4: Action discretization: all actions variables were binned into seven bins. Each combination of bins for all variables was then mapped to a single action, resulting in a total of $7^3 = 343$ discrete actions.

C.2 Significance Tests

		PHWIS		PHWDR	
Algorithm	Compliance	Statistic	p-value	Statistic	p-value
IL	Unconstrained	209.0	0.999	16.0	0.000
	Policy	107.0	0.536	16.0	0.000
QL_D	Unconstrained	–	–	210.0	1.0
	Policy	0.0	0.125	210.0	1.0
	Q-function	–	–	210.0	1.0
QL_S	Unconstrained	93.0	0.337	16.0	0.000
	Policy	80.0	0.184	14.0	0.000
	Q-function	128.0	0.806	25.0	0.000

Table C.5: One-tailed significance test results for the listed policy having a *lower* mean expected return than observed in the test set obtained with Wilcoxon's signed-rank test. Results for QL_D Unconstrained and QL_D Q-function are missing due to an expected sample size of zero.

Appendix D

In this appendix, details on the experiments conducted in Chapter 8 are described.

D.1 Details Experimental Setup

The implementation of the environments and the 'sketch' baseline used in all experiments is the same as the one described by Andreas *et al.*. We include a list of all parameters and a description of the setup and the environments here for completeness. Instructions (see below) were defined in LTL and then converted into FST meta-controllers with the ltlsynt tool by [236] and then implemented in Python. In all of our experiments, we implement each policy as a feedforward neural network with ReLU activations and critics as a linear function of the state. Both are optimized with the RMSProp optimizer. Table D.1 lists all (hyper)parameters used.

Parameter	Value(s)	Description
step size	0.001	RMSProp optimization step size.
γ	0.9	Parameter to balance immediate and long-term rewards.
D	2000	Training algorithm batch size.
ρ	$\{0, 0.1\}$	Intrinsic or shaping rewards.
seeds	$\{0, 1, 2, 3, 4\}$	Initialization of the pseudo-random number generator.

Table D.1: Parameters used in all experiments.

D.2 Craft Environment

The deterministic 'craft' environment is a 10×10 grid in which the agent senses the (x, y) position of locations of interest such as resources and workshops, relative to its own location. The state representation for this environment is a vector of dimensionality 1075, consisting of indicator parameters for each possible item in the agent inventory, indicator parameters for the position of locations of interest relative to the agents position and indicator parameters for the direction the agent is facing. The action space is defined as $\{\text{up}, \text{down}, \text{left}, \text{right}, \text{use}\}$ where the first four always move the agent in the particular direction in the grid. Options are terminated based on instructions or after fifteen time steps and episodes are terminated after 100 time steps.

Task	LTL specification
Plank	$((\texttt{get-wood} \wedge \neg\texttt{w0})\mathbf{W}(\texttt{wood} \vee \texttt{plank})) \wedge (\mathbf{F}\texttt{wood} \implies \mathbf{F}(\texttt{w0}\mathbf{W}\texttt{plank}))$
Stick	$((\texttt{get-wood} \wedge \neg\texttt{w1})\mathbf{W}(\texttt{wood} \vee \texttt{stick})) \wedge (\mathbf{F}\texttt{wood} \implies \mathbf{F}(\texttt{w1}\mathbf{W}\texttt{stick}))$
Cloth	$((\texttt{get-grass} \wedge \neg\texttt{w2})\mathbf{W}(\texttt{grass} \vee \texttt{cloth})) \wedge (\mathbf{F}\texttt{grass} \implies \mathbf{F}(\texttt{w2}\mathbf{W}\texttt{cloth}))$
Rope	$((\texttt{get-grass} \wedge \neg\texttt{w0})\mathbf{W}(\texttt{grass} \vee \texttt{rope})) \wedge (\mathbf{F}\texttt{grass} \implies \mathbf{F}(\texttt{w0}\mathbf{W}\texttt{rope}))$
Bridge	$\mathbf{G}(\neg(\texttt{w2} \wedge \texttt{get-grass}) \wedge \neg(\texttt{w2} \wedge \texttt{get-iron})) \wedge (\texttt{get-wood}\mathbf{W}\texttt{wood}) \wedge (\texttt{get-iron}\mathbf{W}\texttt{iron}) \wedge ((\mathbf{F}\texttt{wood} \wedge \mathbf{F}\texttt{iron}) \implies \mathbf{F}(\texttt{w2}\mathbf{W}\texttt{bridge})$
Bed	$\mathbf{G}(\neg(\texttt{w1} \wedge \texttt{get-wood}) \wedge \neg(\texttt{w1} \wedge \texttt{get-grass}) \wedge \neg(\texttt{w1} \wedge \texttt{w0})) \wedge (\texttt{get-grass}\mathbf{W}\texttt{grass}) \wedge (\texttt{get-wood}\mathbf{W}\texttt{wood}) \wedge (\mathbf{F}(\texttt{wood}) \implies (\texttt{w0}\mathbf{W}\texttt{plank})) \wedge ((\mathbf{F}\texttt{wood} \wedge \mathbf{F}\texttt{plank} \wedge \mathbf{F}\texttt{grass}) \implies \mathbf{F}(\texttt{w1}\mathbf{W}\texttt{bed})$
Axe	$\mathbf{G}(\neg(\texttt{w1} \wedge \texttt{get-wood}) \wedge \neg(\texttt{w1} \wedge \texttt{get-iron}) \wedge \neg(\texttt{w1} \wedge \texttt{w0})) \wedge (\texttt{get-iron}\mathbf{W}\texttt{iron}) \wedge (\texttt{get-wood}\mathbf{W}\texttt{wood}) \wedge (\mathbf{F}(\texttt{wood}) \implies (\texttt{w1}\mathbf{W}\texttt{stick})) \wedge ((\mathbf{F}\texttt{wood} \wedge \mathbf{F}\texttt{stick} \wedge \mathbf{F}\texttt{iron}) \implies \mathbf{F}(\texttt{w0}\mathbf{W}\texttt{axe})$
Shears	$(\texttt{get-iron}\mathbf{W}\texttt{iron}) \wedge (\texttt{get-wood}\mathbf{W}\texttt{wood}) \wedge (\mathbf{F}(\texttt{wood}) \implies (\texttt{w1}\mathbf{W}\texttt{stick})) \wedge ((\mathbf{F}\texttt{wood} \wedge \mathbf{F}\texttt{stick} \wedge \mathbf{F}\texttt{iron}) \implies \mathbf{F}(\texttt{w1}\mathbf{W}\texttt{shears})$
Gold	specification 'Bridge' + $\wedge((\mathbf{F}\texttt{wood} \wedge \mathbf{F}\texttt{iron} \wedge \mathbf{F}\texttt{bridge}) \implies \mathbf{F}(\texttt{get-gold}\mathbf{W}\texttt{gold})$
Gem	specification 'Axe' + $\wedge((\mathbf{F}\texttt{wood} \wedge \mathbf{F}\texttt{stick} \wedge \mathbf{F}\texttt{axe}) \implies \mathbf{F}(\texttt{get-gem}\mathbf{W}\texttt{gem})$

Table D.2: Curriculum of tasks and nondeterministic specifications in the 'craft' environment where $AP^I = \{\texttt{axe}, \texttt{bed}, \texttt{bridge}, \texttt{cloth}, \texttt{door}, \texttt{gem}, \texttt{gold}, \texttt{grass}, \texttt{iron}, \texttt{plank}, \texttt{rope}, \texttt{shears}, \texttt{stick}, \texttt{wood}\}$ each referring to having an item in the agent inventory and behaviors $AP^O = \{\texttt{get-iron}, \texttt{get-wood}, \texttt{get-grass}, \texttt{get-gold}, \texttt{get-gem}, \texttt{w0}, \texttt{w1}, \texttt{w2}\}$ where the latter refer to using three different workshops. These specifications were made deterministic by a total order over all available behaviors.

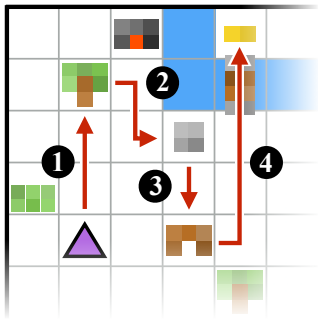


Figure D.1: Visualization of craft world for the 'Gold' task. This task consists of executing (1) get-wood, (2) get-iron, (3) w0, (4) get-gold. The labelling in this world consists of whether an item such as 'wood' is present in the agent inventory. This can be encoded into an LTL specification with vocabulary $AP^I = \{\texttt{wood}, \texttt{iron}, \texttt{bridge}, \texttt{gold}\}$ to describe the environment state and options $AP^O = \{\texttt{get-wood}, \texttt{get-iron}, \texttt{w0}, \texttt{get-gold}\}$.

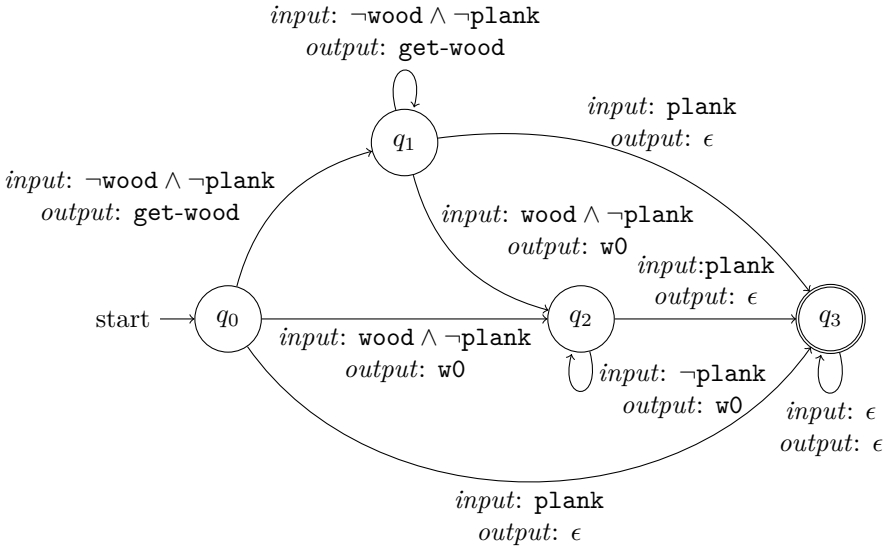


Figure D.2: FST for the 'plank' task generated with the 'ltlsynt' tool of the Spot package by Michaud *et al.*. This specification has input alphabet AP^I : $\{\texttt{wood}, \texttt{plank}\}$ and output alphabet AP^O : $\{\texttt{get-wood}, \texttt{w0}\}$. Negative outputs such as ¬`get-wood` have been omitted in this representation for legibility whereas negative inputs have been included only where necessary to differentiate between available edges. For example, `wood` is not differentiating for any edges leaving q_2. Edges incoming to the terminal node q_3 produce no output: these are only visited if a plank is present in the agent inventory, i.e. upon completion of the task.

D.3 Maze Environment

The 'maze' environment, of which an example is depicted in Figure D.3, is a grid environment of varying size. The environment consists of various adjacent rooms. The agent is placed in one of these rooms and is tasked with reaching a particular other room, possibly by traversing some intermediate rooms. Some rooms are connected by doors, which can be open or locked. Locked doors can be opened by acquiring a key to that particular door and using it on the lock. These keys are placed in a position that is reachable for the agent. The agent senses keys, locked doors and open doors in all cardinal directions and cannot sense through walls. The state representation consists of a vector describing the distance to rooms and keys in all cardinal directions, i.e. it is of dimensionality 12. The action space is defined as $\{\texttt{up}, \texttt{down}, \texttt{left}, \texttt{right}, \texttt{key}\}$ where the first four always move the agent in the particular direction in the grid. Options are terminated based on instructions or after fifteen time steps and episodes are terminated after 100 time steps.

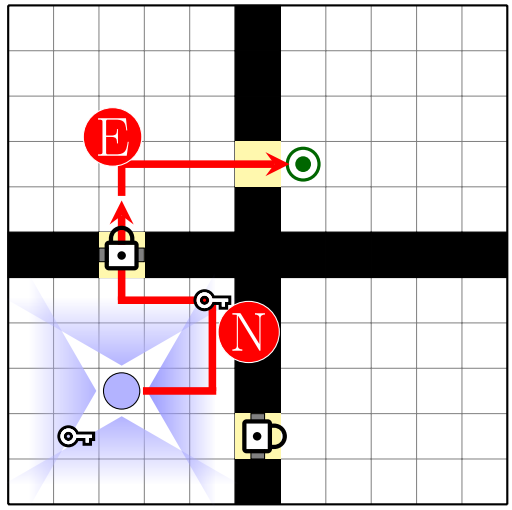


Figure D.3: Visualization of a sample maze environment for the task 'north-east'. Door states, i.e. states such that $L(s) = 1^{\texttt{door}}$, are visualized as a yellow cell All other states are labelled as 'not door' states and visualized as a white cell The agent (blue circle)senses its environment in four cardinal directions with three sensors per direction: the first detects open doors, the second senses locked doors and the last detects keys. The bottom left room contains two keys that look identical to the agent. One unlocks the door to the bottom right room, which need not be visited. The other unlocks the door to the top left and needs to be picked up to reach the target. Arrows denote options associated with 'north' and 'east' respectively for the specification $(\texttt{door}\mathbf{R}\texttt{north}) \wedge (\texttt{door} \implies \mathbf{XG}\texttt{east})$.

Task	LTL specification
WW	$\mathbf{F}\text{door} \wedge \mathbf{G}\text{west}$
WS	$(\text{door}\mathbf{R}\text{west}) \wedge (\text{door} \implies \mathbf{XG}\text{south})$
EE	$(\text{door}\mathbf{R}\text{east}) \wedge (\text{door} \implies \mathbf{XG}\text{south})$
NW	$(\text{door}\mathbf{R}\text{north}) \wedge (\text{door} \implies \mathbf{XG}\text{west})$
NE	$(\text{door}\mathbf{R}\text{north}) \wedge (\text{door} \implies \mathbf{XG}\text{east})$
NEN	$(\text{door}\mathbf{R}(\text{north} \wedge \neg\text{east})) \wedge (\mathbf{F}(\text{door}) \implies (\mathbf{F}((\text{door}\mathbf{R}\text{east}) \wedge \mathbf{F}\text{door} \implies \mathbf{XG}\text{north})))$
SEN	$(\text{door}\mathbf{R}(\text{south} \wedge \neg\text{east})) \wedge (\mathbf{F}(\text{door}) \implies (\mathbf{F}((\text{door}\mathbf{R}\text{east}) \wedge \mathbf{F}\text{door} \implies \mathbf{XG}\text{north})))$
WNE	$(\text{door}\mathbf{R}(\text{west} \wedge \neg\text{north})) \wedge (\mathbf{F}(\text{door}) \implies (\mathbf{F}((\text{door}\mathbf{R}\text{north}) \wedge \mathbf{F}\text{door} \implies \mathbf{XG}\text{east})))$
WWS	$((\mathbf{F}(\text{door} \wedge \mathbf{X}(\mathbf{F}\text{door}))\mathbf{R}(\text{west} \wedge \mathbf{X}\text{west})) \wedge ((\mathbf{F}(\text{door} \wedge \mathbf{X}(\mathbf{F}(\text{door})))) \implies \mathbf{FG}(\text{south})) \wedge \mathbf{G}(\neg(\text{south} \wedge \text{west}))$
ESS	$(\text{door}\mathbf{R}\text{east}) \wedge \mathbf{F}(\mathbf{F}\text{door} \implies \mathbf{XG}(\text{south} \wedge \neg\text{east}))$

Table D.3: Curriculum of tasks and specifications in the 'maze' environment where $AP^I = \{\texttt{door}\}$ and $AP^O = \{\text{west}, \text{east}, \text{north}, \text{south}\}$ each referring to a room to move to and not (!) primitive actions. N=North, W=West, S=South and E=East.

D.4 Results per task

The results in the main document are aggregated across tasks of different complexity. To highlight where difference in performance comes from, we split down the performance per task for both environments in Figures D.4 and D.5 and Tables D.5 and D.4.

ρ			WW	WS	ES	NW	NEN	ESS	SEN	WWS	WNE
h	0	D								0.94	0.15
		Sk								0.85	0.14
	.1	D				N/A				0.49	0.08
		Sk								0.87	0.17
i	0	D	0.98	0.94	0.30	0.93	0.08	0.07	0.03	0.91	0.84
		Sk	0.83	0.00	0.01	0.02	0.00	0.00	0.03	0.00	0.00

Table D.4: Maze environment zero-shot task completion rates for 1K evaluations, averaged over 5 random seeds. Each task consist of a sequence of rooms to reach in the cardinal directions 'North', 'South', 'West', 'East'. The task 'WNE', for example, consists of moving one room 'West', one room 'North' and one room 'East' in that order. D=deterministic, Sk=sketch, h=holdout and i=isolation

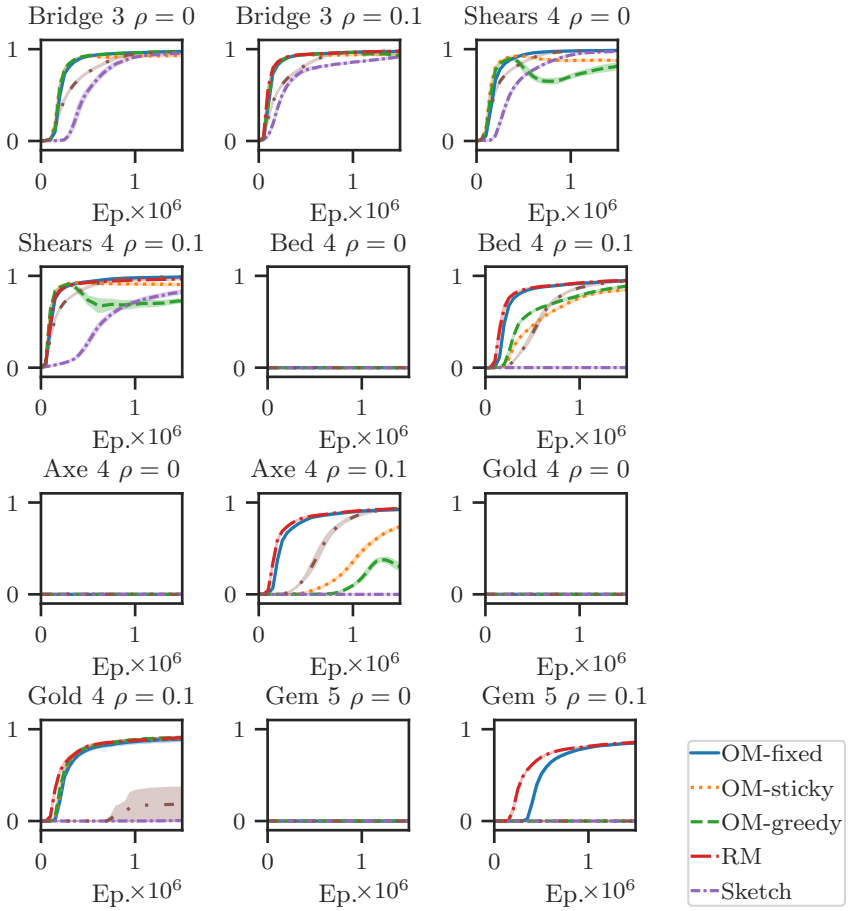


Figure D.4: Total cumulative reward on craft world per task. Titles indicate the task, the number of options and whether shaping was applied ($\rho = 0.1$) or not ($\rho = 0$). The tasks 'bed', 'axe', 'gold' and 'gem' cannot be learned in the single-task setting if reward shaping is not applied as evidenced by graphs in the first and third columns. Reward shaping with our framework allows the learner to solve these hard problems. Additionally, the version of our framework with deterministic options and shaping (bottom right) is the only solution that learns to solve the 'gem' task.

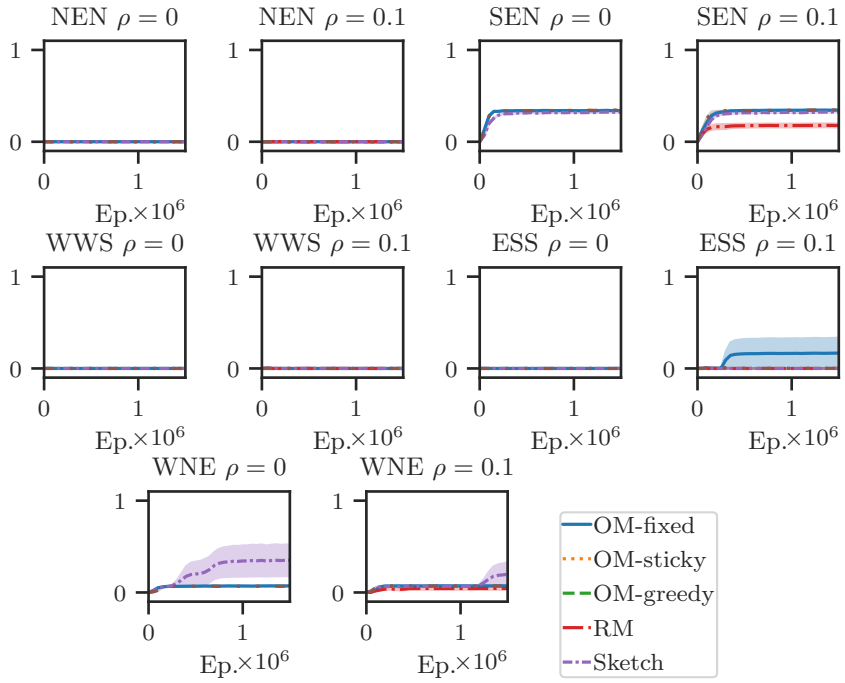


Figure D.5: Task completion on maze world per task. Each task consist of a sequence of rooms to reach in the cardinal directions 'North', 'South', 'West', 'East'. The task 'NEN', for example, consists of moving one room 'North', one room 'East' and one room 'North' in that order.

A

	ρ		Plank	Stick	Rope	Cloth	Bridge	Shears	Bed	Axe
h	0	D							0.94	0.83
		G				N/A			0.19	0.06
		St							0.29	0.10
		Sk							0.09	0.00
	.1	D							0.87	0.84
		G				N/A			0.2	0.06
		St							0.3	0.14
		Sk							0.06	0.00
i	0	D	0.96	0.80	0.97	0.97	0.94	0.91	0.91	0.76
		G	0.96	0.80	0.97	0.97	0.94	0.94	0.15	0.01
		St	0.96	0.80	0.97	0.97	0.95	0.95	0.26	0.11
		Sk	0.00	0.00	0.17	0.00	0.03	0.02	0.00	0.00

Table D.5: Craft environment zero-shot results per task for 1K evaluations, averaged over 5 random seeds. D=Deterministic, G=Greedy, St=sticky, Sk=Sketch, h=holdout and i=isolation.

A

List of Tables

List of Figures

F

Bibliography

[1] N. Abe, N. Verma, C. Apte and R. Schroko. 'Cross channel optimized marketing by reinforcement learning'. In: *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04* (2004).

[2] G. Abowd, A. Dey, P. Brown, N. Davies, M. Smith and P. Steggles. 'Towards a better understanding of context and context-awareness'. In: *Handheld and ubiquitous computing*. Springer. 1999, page 319.

[3] G. Adomavicius and A. Tuzhilin. 'Context-aware recommender systems'. In: *Recommender systems handbook*. Springer, 2011, pages 217–253.

[4] S. Ahrndt, M. Lützenberger and S. M. Prochnow. 'Using Personality Models as Prior Knowledge to Accelerate Learning About Stress-Coping Preferences: (Demonstration)'. In: *AAMAS*. 2016.

[5] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum and U. Topcu. 'Safe reinforcement learning via shielding'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Volume 32. 2018.

[6] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman and D. Mané. 'Concrete problems in AI safety'. In: *arXiv preprint arXiv:1606.06565* (2016).

[7] A. Ando, R. Masumura, H. Kamiyama, S. Kobashikawa and Y. Aono. 'Hierarchical LSTMs with Joint Learning for Estimating Customer Satisfaction from Contact Center Calls'. In: *Interspeech 2017*. ISCA, Aug. 2017.

[8] G. Andrade, G. Ramalho, H. Santana and V. Corruble. 'Automatic computer game balancing'. In: *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems - AAMAS '05* (2005).

[9] J. Andreas, D. Klein and S. Levine. 'Modular multitask reinforcement learning with policy sketches'. In: *International Conference on Machine Learning*. PMLR. 2017, pages 166–175.

[10] G. Antoniou and F. Van Harmelen. *A semantic web primer*. MIT press, 2004.

[11] J. April, M. Better, F. W. Glover, J. P. Kelly and G. A. Kochenberger. 'Ensuring Workforce Readiness with OptForce'. unpublished manuscript retrieved from opttek.com. 2013.

[12] M. G. Aspinall and R. G. Hamermesh. 'Realizing the promise of personalized medicine'. In: *Harvard business review* 85.10 (2007), page 108.

[13] L. E. Asri, H. Schulz, S. Sharma, J. Zumer, J. Harris, E. Fine, R. Mehrotra and K. Suleman. 'Frames: A Corpus for Adding Memory to Goal-Oriented Dialogue Systems'. In: *arXiv:1704.00057 [cs]* (Mar. 2017). arXiv: 1704.00057. (Visited on 29/04/2019).

[14] A. Atrash and J. Pineau. 'A bayesian reinforcement learning approach for customizing human-robot interfaces'. In: *Proceedingsc of the 13th international conference on Intelligent user interfaces - IUI '09* (2008).

[15] P. Auer, N. Cesa-Bianchi and P. Fischer. 'Finite-time analysis of the multiarmed bandit problem'. In: *Machine learning* 47.2-3 (2002), pages 235–256.

[16] S. Ávila-Sansores, F. Orihuela-Espina and L. Enrique-Sucar. 'Patient tailored virtual rehabilitation'. In: *Converging Clinical and Engineering Research on Neurorehabilitation*. Springer, 2013, pages 879–883.

[17] N. F. Awad and M. S. Krishnan. 'The personalization privacy paradox: an empirical evaluation of information transparency and the willingness to be profiled online for personalization'. In: *MIS quarterly* (2006), pages 13–28.

[18] N. Bagdure and B. Ambudkar. 'Reducing Delay during Vertical Handover'. In: *2015 International Conference on Computing Communication Control and Automation* (2015).

[19] C. Baier and J.-P. Katoen. *Principles of model checking*. MIT press, 2008.

[20] J. Bang, H. Noh, Y. Kim and G. G. Lee. 'Example-based chat-oriented dialogue system with personalized long-term memory'. In: *2015 International Conference on Big Data and Smart Computing (BigComp)*. IEEE. 2015, pages 238–243.

[21] A. Baniya, S. Herrmann, Q. Qiao and H. Lu. 'Adaptive Interventions Treatment Modelling and Regimen Optimization Using Sequential Multiple Assignment Randomized Trials (SMART) and Q-learning'. In: *IIE Annual Conference. Proceedings*. Institute of Industrial and Systems Engineers (IISE). 2017, pages 1187–1192.

[22] T. Banyai, C. Landschutzer and A. Banyai. 'Markov-Chain Simulation-Based Analysis of Human Resource Structure: How Staff Deployment and Staffing Affect Sustainable Human Resource Strategy'. In: *Sustainability* 10.10 (2018).

[23] A. Barto, P. Thomas and R. Sutton. 'Some Recent Applications of Reinforcement Learning'. In: (2017).

[24] A. L. C. Bazzan. 'Synergies between evolutionary computation and multiagent reinforcement learning'. In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion on - GECCO '17* (2017).

[25] M. Bellemare, S. Srinivasan, G. Ostrovski, T. Schaul, D. Saxton and R. Munos. 'Unifying count-based exploration and intrinsic motivation'. In: *Advances in neural information processing systems* 29 (2016), pages 1471–1479.

[26] M. G. Bellemare, S. Candido, P. S. Castro, J. Gong, M. C. Machado, S. Moitra, S. S. Ponda and Z. Wang. 'Autonomous navigation of stratospheric balloons using reinforcement learning'. In: *Nature* 588.7836 (2020), pages 77–82.

[27] M. G. Bellemare, Y. Naddaf, J. Veness and M. Bowling. 'The arcade learning environment: An evaluation platform for general agents'. In: *Journal of Artificial Intelligence Research* 47 (2013), pages 253–279.

[28] R. E. Bellman. *Adaptive control processes: a guided tour*. Volume 2045. Princeton university press, 2015.

[29] Y. Bengio, J. Louradour, R. Collobert and J. Weston. 'Curriculum learning'. In: *ICML*. Volume 26. 2009, pages 41–48.

[30] A. Bergmann, K. C. Hall and S. M. Ross. *Language files: Materials for an introduction to language and linguistics*. Ohio State University Press, 2007.

[31] S. Bhulai, G. Koole and A. Pot. 'Simple methods for shift scheduling in multiskill call centers'. In: *Manufacturing & Service Operations Management* 10.3 (2008), pages 411–420.

[32] H. Bi, O. J. Akinwande and E. Gelenbe. 'Emergency Navigation in Confined Spaces Using Dynamic Grouping'. In: *2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies* (2015).

[33] G. Biegel and V. Cahill. 'A framework for developing mobile, context-aware applications'. In: *Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on*. IEEE. 2004, pages 361–365.

[34] R. Bloem, B. Könighofer, R. Könighofer and C. Wang. 'Shield synthesis'. In: *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer. 2015, pages 533–548.

[35] A. Bodas, B. Upadhyay, C. Nadiger and S. Abdelhak. 'Reinforcement learning for game personalization on edge devices'. In: *2018 International Conference on Information and Computer Technologies (ICICT)* (2018).

[36] D. Bouneffouf, A. Bouzeghoub and A. L. Gançarski. 'Hybrid-ϵ-greedy for Mobile Context-Aware Recommender System'. In: *Lecture Notes in Computer Science* (2012), pages 468–479.

[37] R. I. Brafman, G. De Giacomo and F. Patrizi. 'LTLf/LDLf Non-Markovian Rewards'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Volume 32. 2018.

[38] J. Bragg, Mausam and D. S. Weld. 'Optimal Testing for Crowd Workers'. In: *AAMAS*. 2016.

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

[40] P. Brusilovski, A. Kobsa and W. Nejdl. *The adaptive web: methods and strategies of web personalization*. Volume 4321. Springer Science & Business Media, 2007.

[41] D. Budgen and P. Brereton. 'Performing systematic literature reviews in software engineering'. In: *Proceedings of the 28th international conference on Software engineering*. ACM. 2006, pages 1051–1052.

[42] A. B. Buduru and S. S. Yau. 'An Effective Approach to Continuous User Authentication for Touch Screen Smart Devices'. In: *2015 IEEE International Conference on Software Quality, Reliability and Security* (2015).

[43] M. Buhrmester, T. Kwang and S. D. Gosling. 'Amazon's Mechanical Turk: A New Source of Inexpensive, Yet High-Quality, Data?' en. In: *Perspectives on Psychological Science* 6.1 (Jan. 2011), pages 3–5. (Visited on 11/04/2019).

[44] Y. Burda, H. Edwards, D. Pathak, A. Storkey, T. Darrell and A. A. Efros. 'Large-Scale Study of Curiosity-Driven Learning'. In: *International Conference on Learning Representations*. 2019.

[45] E. K. Burke, P. De Causmaecker, G. V. Berghe and H. Van Landeghem. 'The state of the art of nurse rostering'. In: *Journal of scheduling* 7.6 (2004), pages 441–499.

[46] A. Burmania, S. Parthasarathy and C. Busso. 'Increasing the Reliability of Crowdsourcing Evaluations Using Online Quality Assessment'. In: *IEEE Transactions on Affective Computing* 7.4 (Oct. 2016), pages 374–388. (Visited on 04/04/2019).

[47] A. Camacho, O. Chen, S. Sanner and S. A. McIlraith. 'Non-markovian rewards expressed in LTL: guiding search via reward shaping'. In: *Tenth Annual Symposium on Combinatorial Search*. 2017.

B

[48] A. Camacho, R. T. Icarte, T. Q. Klassen, R. A. Valenzano and S. A. McIlraith. 'LTL and Beyond: Formal Languages for Reward Function Specification in Reinforcement Learning'. In: *Proceedings of the 28th Joint Conference on Artificial Intelligence*. Volume 19. 2019, pages 6065–6073.

[49] I. Casanueva, T. Hain, H. Christensen, R. Marxer and P. Green. 'Knowledge transfer between speakers for personalised dialogue management'. In: *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. 2015, pages 12–21.

[50] I. Casanueva, P. Budzianowski, P.-H. Su, N. Mrkšić, T.-H. Wen, S. Ultes, L. Rojas-Barahona, S. Young and M. Gašić. 'A Benchmarking Environment for Reinforcement Learning Based Task Oriented Dialogue Management'. In: *Deep Reinforcement Learning Symposium, 31st Conference on Neural Information Processing Systems*. 2017.

[51] A. Castro-Gonzalez, F. Amirabdollahian, D. Polani, M. Malfaz and M. A. Salichs. 'Robot self-preservation and adaptation to user preferences in game play, a preliminary study'. In: *2011 IEEE International Conference on Robotics and Biomimetics* (2011).

[52] L. Cella. 'Modelling User Behaviors with Evolving Users and Catalogs of Evolving Items'. In: *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization - UMAP '17* (2017).

[53] B. Chakraborty and S. A. Murphy. 'Dynamic Treatment Regimes'. In: *Annual Review of Statistics and Its Application* 1.1 (2014), pages 447–464.

[54] J. Chan and G. Nejat. 'A learning-based control architecture for an assistive robot providing social engagement during cognitively stimulating activities'. In: *2011 IEEE International Conference on Robotics and Automation* (2011).

[55] R. K. Chellappa and R. G. Sin. 'Personalization versus privacy: An empirical examination of the online consumer's dilemma'. In: *Information technology and management* 6.2-3 (2005), pages 181–202.

[56] J. Chen and Z. Yang. 'A learning multi-agent system for personalized information fiftering'. In: *Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint* (2003).

[57] S. Chen, X. Qiu, X. Tan, Z. Fang and Y. Jin. 'A model-based hybrid soft actor-critic deep reinforcement learning algorithm for optimal ventilator settings'. In: *Information Sciences* 611 (2022), pages 47–64.

[58] X. Chen, Y. Zhai, C. Lu, J. Gong and G. Wang. 'A learning model for personalized adaptive cruise control'. In: *2017 IEEE Intelligent Vehicles Symposium (IV)* (2017).

[59] Z. Cheng, Q. Zhao, F. Wang, Y. Jiang, L. Xia and J. Ding. 'Satisfaction based Q-learning for integrated lighting and blind control'. In: *Energy and Buildings* 127 (2016), pages 43–55.

[60] C.-Y. Chi, R. T.-H. Tsai, J.-Y. Lai and J. Y.-j. Hsu. 'A Reinforcement Learning Approach to Emotion-based Automatic Playlist Generation'. In: *2010 International Conference on Technologies and Applications of Artificial Intelligence* (2010).

[61] M. Chi, K. VanLehn, D. Litman and P. Jordan. 'Inducing Effective Pedagogical Strategies Using Learning Context Features'. In: *Lecture Notes in Computer Science* (2010), pages 147–158.

[62] Y.-S. Chiang, T.-S. Chu, C. D. Lim, T.-Y. Wu, S.-H. Tseng and L.-C. Fu. 'Personalizing robot behavior for interruption in social human-robot interaction'. In: *2014 IEEE International Workshop on Advanced Robotics and its Social Impacts* (2014).

[63] W. Chu, L. Li, L. Reyzin and R. Schapire. 'Contextual bandits with linear payoff functions'. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. 2011, pages 208–214.

[64] M. Claeys, S. Latre, J. Famaey and F. De Turck. 'Design and Evaluation of a Self-Learning HTTP Adaptive Video Streaming Client'. In: *IEEE Communications Letters* 18.4 (2014), pages 716–719.

[65] J. Cohen. 'A coefficient of agreement for nominal scales'. In: *Educational and psychological measurement* 20.1 (1960), pages 37–46.

[66] A. Cotten. *Seven steps of effective workforce planning*. IBM Center for the Business of Government, 2007.

[67] G. Da Silveira, D. Borenstein and F. S. Fogliatto. 'Mass customization: Literature review and research directions'. In: *International journal of production economics* 72.1 (2001), pages 1–13.

[68] M. Daltayanni, C. Wang and R. Akella. 'A Fast Interactive Search System for Healthcare Services'. In: *2012 Annual SRII Global Conference* (2012).

[69] E. Daskalaki, P. Diem and S. G. Mougiakakou. 'An Actor–Critic based controller for glucose regulation in type 1 diabetes'. In: *Computer Methods and Programs in Biomedicine* 109.2 (2013), pages 116–125.

[70] E. Daskalaki, P. Diem and S. G. Mougiakakou. 'Model-Free Machine Learning in Biomedicine: Feasibility Study in Type 1 Diabetes'. In: *PLOS ONE* 11.7 (2016). Edited by K. Maedler, e0158722.

[71] E. Daskalaki, P. Diem and S. G. Mougiakakou. 'Personalized tuning of a reinforcement learning control algorithm for glucose regulation'. In: *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (2013).

[72] M. Davis, Y. Lu, M. Sharma, M. Squillante and B. Zhang. 'Stochastic optimization models for workforce planning, operations, and risk management'. In: *Service Science* 10.1 (2018), pages 40–57.

[73] T. De Feyter, M. Guerry et al. 'Optimizing cost-effectiveness in a stochastic Markov manpower planning system under control by recruitment'. In: *Annals of Operations Research* 253.1 (2017), pages 117–131.

[74] G. De Giacomo, M. Favorito, L. Iocchi and F. Patrizi. 'Imitation learning over heterogeneous agents with restraining bolts'. In: *Proceedings of the International Conference on Automated Planning and Scheduling*. Volume 30. 2020, pages 517–521.

[75] G. De Giacomo, L. Iocchi, M. Favorito and F. Patrizi. 'Foundations for restraining bolts: Reinforcement learning with LTLf/LDLf restraining specifications'. In: *Proceedings of the International Conference on Automated Planning and Scheduling*. Volume 29. 2019, pages 128–136.

[76] M. De Paula, G. G. Acosta and E. C. Martínez. 'On-line policy learning and adaptation for real-time personalization of an artificial pancreas'. In: *Expert Systems with Applications* 42.4 (2015), pages 2234–2255.

[77] M. De Paula, L. O. Ávila and E. C. Martínez. 'Controlling blood glucose variability under uncertainty using reinforcement learning and Gaussian processes'. In: *Applied Soft Computing* 35 (2015), pages 310–332.

[78] J. Degrave, F. Felici, J. Buchli, M. Neunert, B. Tracey, F. Carpanese, T. Ewalds, R. Hafner, A. Abdolmaleki, D. de Las Casas et al. 'Magnetic control of tokamak plasmas through deep reinforcement learning'. In: *Nature* 602.7897 (2022), pages 414–419.

[79] K. Deng, J. Pineau and S. Murphy. 'Active learning for personalizing treatment'. In: *2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)* (2011).

[80] F. den Hengst, E. Grua, A. el Hassouni and M. Hoogendoorn. *Release of the systematic literature review into Reinforcement Learning for personalization*. Version v0.0.1. Zenodo, Jan. 2020.

[81] K. Dentler, A. ten Teije, R. Cornet and N. de Keizer. 'Towards the automated calculation of clinical quality indicators'. In: *Knowledge Representation for Health-Care: AIME 2011 Workshop KR4HC 2011, Bled, Slovenia, July 2-6, 2011*. Springer. 2012, pages 51–64.

[82] A. A. Deshmukh, Ü. Dogan and C. Scott. 'Multi-Task Learning for Contextual Bandits'. In: *NIPS*. 2017.

[83] T. Dietterich. 'Hierarchical reinforcement learning with the MAXQ value function decomposition'. In: *JAIR* 13 (2000), pages 227–303.

[84] S. Doroudi, P. S. Thomas and E. Brunskill. 'Importance sampling for fair policy selection'. In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 2018, pages 5239–5243.

[85] Y. Duan, X. Chen, R. Houthooft, J. Schulman and P. Abbeel. 'Benchmarking deep reinforcement learning for continuous control'. In: *International Conference on Machine Learning*. 2016, pages 1329–1338.

[86] G. Dulac-Arnold, N. Levine, D. J. Mankowitz, J. Li, C. Paduraru, S. Gowal and T. Hester. 'Challenges of real-world reinforcement learning: definitions, benchmarks and analysis'. In: *Machine Learning* 110.9 (2021), pages 2419–2468.

[87] A. Durand and J. Pineau. 'Adaptive treatment allocation using subsampled gaussian processes'. In: *2015 AAAI Fall Symposium Series*. 2015.

[88] C. Dwork. 'Differential privacy: A survey of results'. In: *International Conference on Theory and Applications of Models of Computation*. Springer. 2008, pages 1–19.

[89] M. El Fouki, N. Aknin and K. E. El. Kadiri. 'Intelligent Adapted e-Learning System based on Deep Reinforcement Learning'. In: *Proceedings of the 2nd International Conference on Computing and Wireless Communication Systems - ICCWCS'17* (2017).

[90] A. El Hassouni, M. Hoogendoorn, A. E. Eiben, M. van Otterlo and V. Muhonen. 'End-to-end Personalization of Digital Health Interventions using Raw Sensor Data with Deep Reinforcement Learning: A comparative study in digital health interventions for behavior change'. In: *2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. IEEE. 2019, pages 258–264.

[91] A. el Hassouni, M. Hoogendoorn, M. van Otterlo and E. Barbaro. 'Personalization of health interventions using cluster-based reinforcement learning'. In: *International Conference on Principles and Practice of Multi-Agent Systems*. Springer. 2018, pages 467–475.

[92] D. Ernst, P. Geurts and L. Wehenkel. 'Tree-based batch mode reinforcement learning'. In: *Journal of Machine Learning Research* 6 (2005).

[93] H. Fan and M. S. Poole. 'What is personalization? Perspectives on the design and implementation of personalization in information systems'. In: *Journal of Organizational Computing and Electronic Commerce* 16.3-4 (2006), pages 179–202.

[94] M. Fatemi, L. El Asri, H. Schulz, J. He and K. Suleman. 'Policy Networks with Two-Stage Training for Dialogue Systems'. In: *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. 2016, pages 101–110.

[95] Y. Fei, Z. Yang, Y. Chen, Z. Wang and Q. Xie. 'Risk-sensitive reinforcement learning: Near-optimal risk-sample tradeoff in regret'. In: *Advances in Neural Information Processing Systems* 33 (2020), pages 22384–22395.

[96] J. Feng, H. Li, M. Huang, S. Liu, W. Ou, Z. Wang and X. Zhu. 'Learning to Collaborate'. In: *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18* (2018).

[97] B. Fernandez-Gauna and M. Grana. 'Recipe tuning by reinforcement learning in the SandS ecosystem'. In: *2014 6th International Conference on Computational Aspects of Social Networks* (2014).

[98] S. M. Fernando, E. Fan, B. Rochwerg, K. E. Burns, L. J. Brochard, D. J. Cook, A. J. Walkey, N. D. Ferguson, C. L. Hough, D. Brodie et al. 'Lung-protective ventilation and associated outcomes and costs among patients receiving invasive mechanical ventilation in the ED'. In: *Chest* 159.2 (2021), pages 606–618.

[99] S. Ferretti, S. Mirri, C. Prandi and P. Salomoni. 'Automatic web content personalization through reinforcement learning'. In: *Journal of Systems and Software* 121 (2016), pages 157–169.

[100] S. Ferretti, S. Mirri, C. Prandi and P. Salomoni. 'Exploiting Reinforcement Learning to Profile Users and Personalize Web Pages'. In: *2014 IEEE 38th International Computer Software and Applications Conference Workshops* (2014).

[101] S. Ferretti, S. Mirri, C. Prandi and P. Salomoni. 'On personalizing Web content through reinforcement learning'. In: *Universal Access in the Information Society* 16.2 (2016), pages 395–410.

[102] S. Ferretti, S. Mirri, C. Prandi and P. Salomoni. 'On personalizing Web content through reinforcement learning'. In: *Universal Access in the Information Society* 16.2 (2017), pages 395–410.

[103] S. Ferretti, S. Mirri, C. Prandi and P. Salomoni. 'User centered and context dependent personalization through experiential transcoding'. In: *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)* (2014).

[104] P. Festor, Y. Jia, A. C. Gordon, A. A. Faisal, I. Habli and M. Komorowski. 'Assuring the safety of AI-based clinical decision support systems: a case study of the AI Clinician for sepsis treatment'. In: *BMJ health & care informatics* 29.1 (2022).

[105] A. Finnerty, P. Kucherbaev, S. Tranquillini and G. Convertino. 'Keep it simple: reward and task design in crowdsourcing'. en. In: *Proceedings of the Biannual Conference of the Italian Chapter of SIGCHI on - CHItaly '13*. Trento, Italy: ACM Press, 2013, pages 1–4. (Visited on 04/04/2019).

[106] J. L. Fleiss. 'Measuring nominal scale agreement among many raters'. In: *Psychological bulletin* 76.5 (1971), page 378.

[107] L. Fournier. 'Learning capabilities for improving automatic transmission control'. In: *Proceedings of the Intelligent Vehicles '94 Symposium* (1994).

[108] J. Fu and U. Topcu. 'Probably approximately correct MDP learning and control with temporal logic constraints'. In: *Proceedings of Robotics: Science and Systems*. 2014.

[109] J. Futoma, M. Hughes and F. Doshi-Velez. 'POPCORN: Partially Observed Prediction Constrained Reinforcement Learning'. In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Edited by S. Chiappa and R. Calandra. Volume 108. Proceedings of Machine Learning Research. PMLR, Aug. 2020, pages 3578–3588.

[110] J. Futoma, M. A. Masood and F. Doshi-Velez. 'Identifying distinct, effective treatments for acute hypotension with SODA-RL: safely optimized diverse accurate reinforcement learning'. In: *AMIA Summits on Translational Science Proceedings* 2020 (2020), page 181.

[111] C. Gaimon and G. Thompson. 'A distributed parameter cohort personnel planning model that uses cross-sectional data'. In: *Management Science* 30.6 (1984), pages 750–764.

[112] A. Y. Gao, W. Barendregt and G. Castellano. 'Personalised human-robot co-adaptation in instructional settings using reinforcement learning'. In: *IVA Workshop on Persuasive Embodied Agents for Behavior Change: PEACH 2017, August 27, Stockholm, Sweden*. 2017.

[113] M. Gaon and R. Brafman. 'Reinforcement Learning with Non-Markovian Rewards'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Volume 34. 2020, pages 3980–3987.

[114] J. Garcıa and F. Fernández. 'A comprehensive survey on safe reinforcement learning'. In: *Journal of Machine Learning Research* 16.1 (2015), pages 1437–1480.

[115] A. Garivier and E. Moulines. 'On upper-confidence bound policies for switching bandit problems'. In: *International Conference on Algorithmic Learning Theory*. Springer. 2011, pages 174–188.

[116] M. Gasic, F. Jurcicek, S. Keizer, F. Mairesse, B. Thomson, K. Yu and S. Young. 'Gaussian processes for fast policy optimisation of pomdp-based dialogue managers'. In: *Proceedings of the SIGDIAL 2010 Conference*. 2010, pages 201–204.

[117] A. Gaweda, M. Muezzinoglu, G. Aronoff, A. Jacobs, J. Zurada and M. Brier. 'Incorporating Prior Knowledge into Q-Learning for Drug Delivery Individualization'. In: *Fourth International Conference on Machine Learning and Applications (ICMLA'05)* (2005).

[118] A. E. Gaweda, M. K. Muezzinoglu, G. R. Aronoff, A. A. Jacobs, J. M. Zurada and M. E. Brier. 'Individualization of pharmacological anemia management using reinforcement learning'. In: *Neural Networks* 18.5 (2005), pages 826–834.

[119] A. E. Gaweda. 'Improving management of Anemia in End Stage Renal Disease using Reinforcement Learning'. In: *2009 International Joint Conference on Neural Networks* (2009).

[120] A. Genevay and R. Laroche. 'Transfer learning for user adaptation in spoken dialogue systems'. In: *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. 2016, pages 975–983.

[121] C. Gentile, S. Li and G. Zappella. 'Online clustering of bandits'. In: *International Conference on Machine Learning*. 2014, pages 757–765.

[122] B. S. Ghahfarokhi and N. Movahhedinia. 'A personalized QoE-aware handover decision based on distributed reinforcement learning'. In: *Wireless Networks* 19.8 (2013), pages 1807–1828.

[123] G. S. Ginsburg and J. J. McCarthy. 'Personalized medicine: revolutionizing drug discovery and patient care'. In: *TRENDS in Biotechnology* 19.12 (2001), pages 491–496.

[124] R. Gligorov. 'Towards Integration of End-User Tags with Professional Annotations'. In: 2010.

[125] D. Glowacka, T. Ruotsalo, K. Konuyshkova, k. Athukorala, S. Kaski and G. Jacucci. 'Directing exploratory search'. In: *Proceedings of the 2013 international conference on Intelligent user interfaces - IUI '13* (2013).

[126] M. H. Göker and C. A. Thompson. 'Personalized conversational case-based recommendation'. In: *European Workshop on Advances in Case-Based Reasoning*. Springer. 2000, pages 99–111.

[127] Y. Goldberg and M. R. Kosorok. 'Q-learning with censored data'. In: *The Annals of Statistics* 40.1 (2012), pages 529–560.

[128] E. C. Goligher, N. D. Ferguson and L. J. Brochard. 'Clinical challenges in mechanical ventilation'. In: *The Lancet* 387.10030 (2016), pages 1856–1866.

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

[130] O. Gottesman, F. Johansson, M. Komorowski, A. Faisal, D. Sontag, F. Doshi-Velez and L. A. Celi. 'Guidelines for reinforcement learning in healthcare'. In: *Nature medicine* 25.1 (2019), pages 16–18.

[131] R. Grinold and R. Stanford. 'Optimal control of a graded manpower system'. In: *Management Science* 20.8 (1974), pages 1201–1216.

[132] M. Grounds and D. Kudenko. 'Combining reinforcement learning with symbolic planning'. In: *AAMAS*. Springer, 2005, pages 75–86.

[133] E. M. Grua and M. Hoogendoorn. 'Exploring Clustering Techniques for Effective Reinforcement Learning based Personalization for Health and Wellbeing'. In: *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE. 2018, pages 813–820.

[134] M. Grzes and D. Kudenko. 'Plan-based reward shaping for reinforcement learning'. In: *International IEEE Conference Intelligent Systems*. Volume 2. IEEE. 2008, pages 10–22.

[135] M. Grześ. 'Reward Shaping in Episodic Reinforcement Learning'. In: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. 2017, pages 565–573.

[136] S. Gu, E. Holly, T. Lillicrap and S. Levine. 'Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates'. In: *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2017, pages 3389–3396.

[137] T. Haarnoja, A. Zhou, P. Abbeel and S. Levine. 'Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor'. In: *International Conference on Machine Learning*. 2018, pages 1856–1865.

[138] B. Hao, X. Ji, Y. Duan, H. Lu, C. Szepesvári and M. Wang. 'Bootstrapping fitted q-evaluation for off-policy inference'. In: *International Conference on Machine Learning*. PMLR. 2021, pages 4074–4084.

[139] F. M. Harper, X. Li, Y. Chen and J. A. Konstan. 'An economic model of user rating in an online recommender system'. In: *Lecture notes in computer science* 3538 (2005), page 307.

[140] C. R. Harris et al. 'Array programming with NumPy'. In: *Nature* 585.7825 (Sept. 2020), pages 357–362.

[141] M. Hasanbeig, A. Abate and D. Kroening. 'Cautious Reinforcement Learning with Logical Constraints'. In: *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*. 2020, pages 483–491.

[142] M. Hasanbeig, N. Y. Jeppu, A. Abate, T. Melham and D. Kroening. 'DeepSynth: Automata Synthesis for Automatic Task Segmentation in Deep Reinforcement Learning'. In: *The Thirty-Fifth {AAAI} Conference on Artificial Intelligence,{AAAI}*. Volume 2. 2021, pages 7647–7656.

[143] S. H. Hashemi, K. Williams, A. El Kholy, I. Zitouni and P. A. Crook. 'Measuring User Satisfaction on Smart Speaker Intelligent Assistants Using Intent Sensitive Query Embeddings'. en. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management - CIKM '18*. Torino, Italy: ACM Press, 2018, pages 1183–1192. (Visited on 04/04/2019).

[144] A. F. Hayes and K. Krippendorff. 'Answering the Call for a Standard Reliability Measure for Coding Data'. en. In: *Communication Methods and Measures* 1.1 (Apr. 2007), pages 77–89. (Visited on 21/05/2019).

[145] J. Heger and T. Voss. 'Dynamically Changing Sequencing Rules with Reinforcement Learning in a Job Shop System With Stochastic Influences'. In: *2020 Winter Simulation Conference (WSC)*. 2020, pages 1608–1618.

[146] J. Hemminghaus and S. Kopp. 'Adaptive Behavior Generation for Child-Robot Interaction'. In: *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction - HRI '18* (2018).

[147] M. Henderson, B. Thomson and J. D. Williams. 'The second dialog state tracking challenge'. In: *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*. 2014, pages 263–272.

[148] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar and D. Silver. 'Rainbow: Combining improvements in deep reinforcement learning'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Volume 32. 2018.

[149] T. Hester and P. Stone. 'Learning and Using Models'. In: *Reinforcement learning*. Edited by M. Wiering and M. Van Otterlo. Volume 12. Springer, 2012, "120".

[150] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband et al. 'Deep q-learning from demonstrations'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Volume 32. 1. 2018.

[151] R. Higashinaka, K. Funakoshi, Y. Kobayashi and M. Inaba. 'The dialogue breakdown detection challenge: Task description, datasets, and evaluation metrics'. In: *LREC*. 2016.

[152] D. N. Hill, H. Nassif, Y. Liu, A. Iyer and S. Vishwanathan. 'An Efficient Bandit Algorithm for Realtime Multivariate Optimization'. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '17* (2017).

[153] T. Hiraoka, G. Neubig, S. Sakti, T. Toda and S. Nakamura. 'Learning cooperative persuasive dialogue policies using framing'. In: *Speech Communication* 84 (2016), pages 83–96.

[154] C. Hori, J. Perez, R. Higashinaka, T. Hori, Y.-L. Boureau, M. Inaba, Y. Tsunomori, T. Takahashi, K. Yoshino and S. Kim. 'Overview of the sixth dialog system technology challenge: DSTC6'. In: *Computer Speech & Language* 55 (2019), pages 1–25.

[155] Z. Huajun, Z. Jin, W. Rui and M. Tan. 'Multi-objective reinforcement learning algorithm and its application in drive system'. In: *2008 34th Annual Conference of IEEE Industrial Electronics* (2008).

[156] S.-l. Huang and F.-r. Lin. 'Designing intelligent sales-agent for online selling'. In: *Proceedings of the 7th international conference on Electronic commerce - ICEC '05* (2005).

[157] R. T. Icarte, T. Klassen, R. Valenzano and S. McIlraith. 'Using reward machines for high-level task specification and decomposition in reinforcement learning'. In: *Proceedings of the 37th International Conference on Machine Learning Conference*. 2018, pages 2107–2116.

[158] R. T. Icarte, T. Q. Klassen, R. Valenzano and S. A. McIlraith. 'Reward machines: Exploiting reward function structure in reinforcement learning'. In: *Journal of Artificial Intelligence Research* 73 (2022), pages 173–208.

[159] E. Ie, C.-w. Hsu, M. Mladenov, V. Jain, S. Narvekar, J. Wang, R. Wu and C. Boutilier. 'RecSim: A Configurable Simulation Platform for Recommender Systems'. In: *arXiv preprint arXiv:1909.04847* (2019).

[160] L. Illanes, X. Yan, R. T. Icarte and S. A. McIlraith. 'Symbolic Plans as High-Level Instructions for Reinforcement Learning'. In: *Proceedings of the International Conference on Automated Planning and Scheduling*. Volume 30. 2020, pages 540–550.

[161] P. Jaillet, G. G. Loke and M. Sim. 'Strategic Workforce Planning Under Uncertainty'. In: *Operations Research* (2021).

[162] S. Jaradat, N. Dokoohaki, M. Matskin and E. Ferrari. 'Trust and privacy correlations in social networks: A deep learning framework'. In: *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (2016).

[163] G. Jawaheer, M. Szomszor and P. Kostkova. 'Comparison of implicit and explicit feedback from an online music recommendation service'. In: *proceedings of the 1st international workshop on information heterogeneity and fusion in recommender systems*. ACM. 2010, pages 47–51.

[164] Y. Jia, J. Burden, T. Lawton and I. Habli. 'Safe reinforcement learning for sepsis treatment'. In: *2020 IEEE International conference on healthcare informatics (ICHI)*. IEEE. 2020, pages 1–7.

[165] N. Jiang and L. Li. 'Doubly Robust Off-policy Value Evaluation for Reinforcement Learning'. In: *International Conference on Machine Learning*. 2016, pages 652–661.

[166] Z. Jin and Z. Huajun. 'Multi-objective reinforcement learning algorithm and its improved convergency method'. In: *2011 6th IEEE Conference on Industrial Electronics and Applications* (2011).

[167] V. Jnitova, S. Elsawah and M. Ryan. 'Review of simulation models in military workforce planning and management context'. In: *The Journal of Defense Modeling and Simulation* 14.4 (2017), pages 447–463.

[168] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi and R. G. Mark. 'MIMIC-III, a freely accessible critical care database'. In: *Scientific data* 3.1 (2016), pages 1–9.

[169] S. Junges, N. Jansen, C. Dehnert, U. Topcu and J.-P. Katoen. 'Safety-constrained reinforcement learning for MDPs'. In: *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer. 2016, pages 130–146.

[170] L. P. Kaelbling. 'Learning to achieve goals'. In: *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, IJCAI-93*. International Joint Conference on Artificial Intelligence. 1993.

[171] J.-D. Kant, G. Ballot and O. Goudet. 'WorkSim: An Agent-Based Model of Labor Markets'. In: *Journal of Artificial Societies and Social Simulation* 23.4 (2020), page 4.

[172] A. Karatzoglou, X. Amatriain, L. Baltrunas and N. Oliver. 'Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering'. In: *Proceedings of the fourth ACM conference on Recommender systems*. ACM. 2010, pages 79–86.

[173] A. A. Kardan and O. R. Speily. 'Smart Lifelong Learning System Based on Q-Learning'. In: *2010 Seventh International Conference on Information Technology: New Generations* (2010).

[174] I. Kastanis and M. Slater. 'Reinforcement learning utilizes proxemics'. In: *ACM Transactions on Applied Perception* 9.1 (2012), pages 1–15.

[175] G. Kazai. 'In Search of Quality in Crowdsourcing for Search Engine Evaluation'. In: *Advances in Information Retrieval*. Edited by P. Clough, C. Foley, C. Gurrin, G. J. F. Jones, W. Kraaij, H. Lee and V. Mudoch. Volume 6611. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pages 165–176. (Visited on 03/04/2019).

[176] D. Kelly. 'Methods for Evaluating Interactive Information Retrieval Systems with Users'. In: *Found. Trends Inf. Retr.* 3.1—2 (Jan. 2009), pages 1–224.

[177] M. K. Khribi, M. Jemni and O. Nasraoui. 'Automatic recommendations for e-learning personalization based on web usage mining techniques and information retrieval'. In: *Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference on*. IEEE. 2008, pages 241–245.

[178] Y. Kim, J. Bang, J. Choi, S. Ryu, S. Koo and G. G. Lee. 'Acquisition and use of long-term memory for personalized dialog systems'. In: *International Workshop on Multimodal Analyses Enabling Artificial Agents in Human-Machine Interaction*. Springer. 2014, pages 78–87.

[179] A. Kittur, J. V. Nickerson, M. Bernstein, E. Gerber, A. Shaw, J. Zimmerman, M. Lease and J. Horton. 'The Future of Crowd Work'. In: *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*. CSCW '13. event-place: San Antonio, Texas, USA. New York, NY, USA: ACM, 2013, pages 1301–1318. (Visited on 11/04/2019).

[180] J. Kober and J. Peters. 'Reinforcement Learning in Robotics: A Survey'. In: *Reinforcement learning*. Edited by M. Wiering and M. Van Otterlo. Volume 12. Springer, 2012, "596–597".

[181] M. Komorowski, L. A. Celi, O. Badawi, A. C. Gordon and A. A. Faisal. 'The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care'. In: *Nature medicine* 24.11 (2018), pages 1716–1720.

[182] V. Konda and J. Tsitsiklis. 'Actor-critic algorithms'. In: *NIPS*. 2000, pages 1008–1014.

[183] A. Kong. 'A note on importance sampling using standardized weights'. In: *University of Chicago, Dept. of Statistics, Tech. Rep* 348 (1992).

[184] B. Könighofer, F. Lorber, N. Jansen and R. Bloem. 'Shield synthesis for reinforcement learning'. In: *International Symposium on Leveraging Applications of Formal Methods*. Springer. 2020, pages 290–306.

[185] I. Koukoutsidis. 'A learning strategy for paging in mobile environments'. In: *5th European Personal Mobile Communications Conference 2003* (2003).

[186] E. F. Krakow et al. 'Tools for the Precision Medicine Era: How to Develop Highly Personalized Treatment Recommendations From Cohort and Registry Data Using Q-Learning'. In: *American Journal of Epidemiology* 186.2 (2017), pages 160–172.

[187] T. L. Lai and H. Robbins. 'Asymptotically efficient adaptive allocation rules'. In: *Advances in applied mathematics* 6.1 (1985), pages 4–22.

[188] A. S. Lan and R. G. Baraniuk. 'A Contextual Bandits Framework for Personalized Learning Action Selection'. In: *EDM*. 2016.

[189] N. Lazic, C. Boutilier, T. Lu, E. Wong, B. Roy, M. Ryu and G. Imwalle. 'Data center cooling using model-predictive control'. In: *Advances in Neural Information Processing Systems* 31 (2018).

[190] H. Le, C. Voloshin and Y. Yue. 'Batch policy learning under constraints'. In: *International Conference on Machine Learning*. PMLR. 2019, pages 3703–3712.

[191] J. Le, A. Edmonds, V. Hester and L. Biewald. 'Ensuring quality in crowdsourced search relevance evaluation : The effects of training question distribution'. In: 2011.

[192] L. Leape, D. Berwick, C. Clancy, J. Conway, P. Gluck, J. Guest, D. Lawrence, J. Morath, D. O'Leary, P. O'Neill et al. 'Transforming healthcare: a safety imperative'. In: *BMJ Quality & Safety* 18.6 (2009), pages 424–428.

[193] G. Lee, S. Bauer, P. Faratin and J. Wroclawski. 'Learning user preferences for wireless services provisioning'. In: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004.* (2004), pages 480–487.

[194] M. Leonetti, L. Iocchi and F. Patrizi. 'Automatic generation and learning of finite-state controllers'. In: *AIMSA*. Springer, 2012, pages 135–144.

[195] M. Leonetti, L. Iocchi and P. Stone. 'A synthesis of automated planning and reinforcement learning for efficient, robust decision-making'. In: *Artificial Intelligence* 241 (2016), pages 103–130.

[196] K. Li and M. Q.-H. Meng. 'Personalizing a Service Robot by Learning Human Habits from Behavioral Footprints'. In: *Engineering* 1.1 (2015), pages 079–084.

[197] L. Li, W. Chu, J. Langford and R. E. Schapire. 'A contextual-bandit approach to personalized news article recommendation'. In: *Proceedings of the 19th international conference on World wide web*. ACM. 2010, pages 661–670.

[198] L. Li, W. Chu, J. Langford and R. E. Schapire. 'A contextual-bandit approach to personalized news article recommendation'. In: *Proceedings of the 19th international conference on World wide web - WWW '10* (2010).

[199] X. Li, C. Vasile and C. Belta. 'Reinforcement learning with temporal logic rewards'. In: *IROS*. IEEE/RSJ. 2017, pages 3834–3839.

[200] Z. Li, J. Kiseleva, M. de Rijke and A. Grotov. 'Towards Learning Reward Functions from User Interactions'. In: *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval - ICTIR '17* (2017).

[201] E. Liebman and P. Stone. 'DJ-MC: A Reinforcement-Learning Agent for Music Playlist Recommendation'. In: *AAMAS*. 2015.

[202] J. Lim, H. Son, D. Lee and D. Lee. 'An MARL-Based Distributed Learning Scheme for Capturing User Preferences in a Smart Environment'. In: *2017 IEEE International Conference on Services Computing (SCC)* (2017).

[203] L.-J. Lin. 'Self-improving reactive agents based on reinforcement learning, planning and teaching'. In: *Machine learning* 8.3-4 (1992), pages 293–321.

[204] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick. 'Microsoft COCO: Common Objects in Context'. In: *Computer Vision – ECCV 2014*. Edited by D. Fleet, T. Pajdla, B. Schiele and T. Tuytelaars. Volume 8693. Cham: Springer International Publishing, 2014, pages 740–755. (Visited on 12/04/2019).

[205] D. J. Litman, M. S. Kearns, S. Singh and M. A. Walker. 'Automatic optimization of dialogue management'. In: *COLING 2000 Volume 1: The 18th International Conference on Computational Linguistics*. Association for Computational Linguistics. 2000, pages 502–508.

[206] Q. Liu, B. Cui, Z. Wei, B. Peng, H. Huang, H. Deng, J. Hao, X. Huang and K.-F. Wong. 'Building personalized simulator for interactive search'. In: *Proceedings of the Twenty-eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*. 2019, pages 5109–5115.

[207] Y. Liu, B. Logan, N. Liu, Z. Xu, J. Tang and Y. Wang. 'Deep Reinforcement Learning for Dynamic Treatment Regimes on Medical Registry Data'. In: *2017 IEEE International Conference on Healthcare Informatics (ICHI)* (2017).

[208] Llorente and S. E. Guerrero. 'Increasing Retrieval Quality in Conversational Recommenders'. In: *IEEE Transactions on Knowledge and Data Engineering* 24.10 (2012), pages 1876–1888.

[209] H. M. Lotfy, S. M. Khamis and M. M. Aboghazalah. 'Multi-agents and learning: Implications for Webusage mining'. In: *Journal of Advanced Research* 7.2 (2016), pages 285–295.

[210] C. Lowery and A. A. Faisal. 'Towards efficient, personalized anesthesia using continuous reinforcement learning for propofol infusion control'. In: *2013 6th International IEEE/EMBS Conference on Neural Engineering (NER)* (2013).

[211] J. Luketina, N. Nardelli, G. Farquhar, J. Foerster, J. Andreas, E. Grefenstette, S. Whiteson and T. Rocktäschel. 'A Survey of Reinforcement Learning Informed by Natural Language'. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, July 2019, pages 6309–6317.

[212] D. Lyu, F. Yang, B. Liu and S. Gustafson. 'SDRL: interpretable and data-efficient deep reinforcement learning leveraging symbolic planning'. In: *AAAI*. Volume 33. 2019, pages 2970–2977.

[213] O. Madani and D. DeCoste. 'Contextual recommender problems'. In: *Proceedings of the 1st international workshop on Utility-based data mining*. ACM. 2005, pages 86–89.

[214] O. Madani and D. DeCoste. 'Contextual recommender problems [extended abstract]'. In: *Proceedings of the 1st international workshop on Utility-based data mining - UBDM '05* (2005).

[215] P. Maes and R. Kozierok. 'Learning interface agents'. In: *AAAI*. Volume 93. 1993, pages 459–465.

[216] T. Mahmood, G. Mujtaba and A. Venturini. 'Dynamic personalization in conversational recommender systems'. In: *Information Systems and e-Business Management* 12.2 (2013), pages 213–238.

[217] T. Mahmood, G. Mujtaba and A. Venturini. 'Dynamic personalization in conversational recommender systems'. In: *Information Systems and e-Business Management* 12.2 (2014), pages 213–238.

[218] T. Mahmood and F. Ricci. 'Learning and adaptivity in interactive recommender systems'. In: *Proceedings of the ninth international conference on Electronic commerce - ICEC '07* (2007).

[219] A. Malpani, B. Ravindran and H. Murthy. 'Personalized Intelligent Tutoring System Using Reinforcement Learning'. In: *FLAIRS Conference*. 2011.

[220] I. Manickam, A. S. Lan and R. G. Baraniuk. 'Contextual multi-armed bandit algorithms for personalized learning action selection'. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2017).

[221] K. N. Martin and I. Arroyo. 'AgentX: Using reinforcement learning to improve the effectiveness of intelligent tutoring systems'. In: *International Conference on Intelligent Tutoring Systems*. Springer. 2004, pages 564–572.

[222] J. D. Martín-Guerrero, F. Gomez, E. Soria-Olivas, J. Schmidhuber, M. Climente-Martí and N. V. Jiménez-Torres. 'A reinforcement learning approach for individualizing erythropoietin dosages in hemodialysis patients'. In: *Expert Systems with Applications* 36.6 (2009), pages 9737–9742.

[223] J. D. Martín-Guerrero, F. Gomez, E. Soria-Olivas, J. Schmidhuber, M. Climente-Martí and N. V. Jiménez-Torres. 'A reinforcement learning approach for individualizing erythropoietin dosages in hemodialysis patients'. In: *Expert Systems with Applications* 36.6 (2009), pages 9737–9742.

[224] J. D. Martín-Guerrero, E. Soria-Olivas, M. Martínez-Sober, A. J. Serrrano-López, R. Magdalena-Benedito and J. Gómez-Sanchis. 'Use of reinforcement learning in two real applications'. In: *European Workshop on Reinforcement Learning*. Springer. 2008, pages 191–204.

[225] W. Mason and D. J. Watts. 'Financial Incentives and the "Performance of Crowds"'. In: *Proceedings of the ACM SIGKDD Workshop on Human Computation*. HCOMP '09. event-place: Paris, France. New York, NY, USA: ACM, 2009, pages 77–85. (Visited on 11/04/2019).

[226] D. Massimo, M. Elahi and F. Ricci. 'Learning User Preferences by Observing User-Items Interactions in an IoT Augmented Space'. In: *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization - UMAP '17* (2017).

[227] K. Masumitsu and T. Echigo. 'Video summarization using reinforcement learning in eigenspace'. In: *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)* (2000).

[228] B. C. May, N. Korda, A. Lee and D. S. Leslie. 'Optimistic Bayesian sampling in contextual-bandit problems'. In: *Journal of Machine Learning Research* 13.Jun (2012), pages 2069–2106.

[229] R. Mazala. 'Infinite Games'. In: *Automata Logics, and Infinite Games: A Guide to Current Research*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pages 23–38.

[230] J. McCarthy, M. L. Minsky, N. Rochester and C. E. Shannon. 'A proposal for the Dartmouth summer research project on artificial intelligence, august 31, 1955'. In: *AI magazine* 27.4 (2006), page 12.

[231] M. F. McTear. 'Spoken dialogue technology: enabling the conversational user interface'. In: *ACM Computing Surveys (CSUR)* 34.1 (2002), pages 90–169.

[232] G. Mealy. 'A method for synthesizing sequential circuits'. In: *The Bell System Technical Journal* 34.5 (1955), pages 1045–1079.

[233] E. Mengelkamp, J. Gärttner and C. Weinhardt. 'Intelligent Agent Strategies for Residential Customers in Local Electricity Markets'. In: *Proceedings of the Ninth International Conference on Future Energy Systems - e-Energy '18* (2018).

[234] E. Mengelkamp and C. Weinhardt. 'Clustering Household Preferences in Local Electricity Markets'. In: *Proceedings of the Ninth International Conference on Future Energy Systems - e-Energy '18* (2018).

[235] N. Merkle and S. Zander. 'Agent-Based Assistance in Ambient Assisted Living Through Reinforcement Learning and Semantic Technologies'. In: *Lecture Notes in Computer Science* (2017), pages 180–188.

[236] T. Michaud and M. Colange. 'Reactive synthesis from LTL specification with Spot'. In: *7th Workshop on Synthesis, SYNT@ CAV*. 2018.

[237] A. Mirhoseini, A. Goldie, M. Yazgan, J. W. Jiang, E. Songhori, S. Wang, Y.-J. Lee, E. Johnson, O. Pathak, A. Nazi et al. 'A graph placement methodology for fast chip design'. In: *Nature* 594.7862 (2021), pages 207–212.

[238] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra and M. Riedmiller. 'Playing Atari With Deep Reinforcement Learning'. In: *NIPS Deep Learning Workshop*. 2013.

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

[240] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski et al. 'Human-level control through deep reinforcement learning'. In: *Nature* 518.7540 (2015), pages 529–533.

[241] K. Mo, Y. Zhang, S. Li, J. Li and Q. Yang. 'Personalizing a Dialogue System With Transfer Reinforcement Learning'. In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

[242] K. Mo, Y. Zhang, S. Li, J. Li and Q. Yang. 'Personalizing a Dialogue System With Transfer Reinforcement Learning'. In: *AAAI*. 2018.

[243] D. Moher, A. Liberati, J. Tetzlaff and D. G. Altman. 'Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement'. In: *Annals of internal medicine* 151.4 (2009), pages 264–269.

[244] O. Moling, L. Baltrunas and F. Ricci. 'Optimal radio channel recommendations with explicit and implicit feedback'. In: *Proceedings of the sixth ACM conference on Recommender systems - RecSys '12* (2012).

[245] A. Moon, T. Kang, H. Kim and H. Kim. 'A Service Recommendation Using Reinforcement Learning for Network-based Robots in Ubiquitous Computing Environments'. In: *RO-MAN 2007 - The 16th IEEE International Symposium on Robot and Human Interactive Communication* (2007).

[246] M. A. Musen and Association for Computing Machinery. *On the Role of User-generated Metadata in Audio Visual Collections*. English. OCLC: 1043841581. New York, NY: ACM, 2011.

[247] S. Muylle, R. Moenaert and M. Despontin. 'The conceptualization and empirical validation of web site user satisfaction'. In: *Information & Management* 41.5 (May 2004), pages 543–560. (Visited on 10/04/2019).

[248] A. Najar and M. Chetouani. 'Reinforcement learning with human advice: a survey'. In: *Frontiers in Robotics and AI* 8 (2021), page 584075.

[249] S. Narvekar, J. Sinapov and P. Stone. 'Autonomous Task Sequencing for Customized Curriculum Design in Reinforcement Learning'. In: *IJCAI*. 2017.

[250] S. Nemati, M. M. Ghassemi and G. D. Clifford. 'Optimal medication dosing from suboptimal clinical examples: A deep reinforcement learning approach'. In: *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (2016).

[251] A. R. D. S. Network. 'Ventilation with lower tidal volumes as compared with traditional tidal volumes for acute lung injury and the acute respiratory distress syndrome'. In: *New England Journal of Medicine* 342.18 (2000), pages 1301–1308.

[252] D. Neumann et al. 'A self-taught artificial agent for multi-physics computational model personalization'. In: *Medical Image Analysis* 34 (2016), pages 52–64.

[253] D. Neumann et al. 'Vito – A Generic Agent for Multi-physics Model Personalization: Application to Heart Modeling'. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (2015), pages 442–449.

[254] A. Y. Ng, D. Harada and S. Russell. 'Policy invariance under reward transformations: Theory and application to reward shaping'. In: *Proceedings of the 16th International Conference on Machine Learning*. 1999, pages 278–287.

[255] N. F. Noy, D. L. McGuinness et al. *Ontology development 101: A guide to creating your first ontology*. Technical report SMI-2001-0880. Stanford Medical Informatics, 2001.

[256] D. Oh and C. L. Tan. 'Making Better Recommendations with Online Profiling Agents'. In: *AI Magazine* 26 (2004), pages 29–40.

[257] P. Ondruska and I. Posner. 'The route not taken: Driver-centric estimation of electric vehicle range'. In: *Twenty-Fourth International Conference on Automated Planning and Scheduling*. 2014.

[258] S. J. Pan and Q. Yang. 'A survey on transfer learning'. In: *IEEE Transactions on knowledge and data engineering* 22.10 (2010), pages 1345–1359.

[259] V. Pant, S. Bhasin and S. Jain. 'Self-learning system for personalized e-learning'. In: *2017 International Conference on Emerging Trends in Computing and Communication Technologies (ICETCCT)* (2017).

[260] R. Parr and S. Russell. 'Reinforcement learning with Rhierarchies of machines'. In: *NIPS* (1998), pages 1043–1049.

[261] R. Passonneau, N. Habash and O. Rambow. 'Inter-annotator Agreement on a Multilingual Semantic Annotation Task'. In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*. 2006.

[262] P. Patompak, S. Jeong, I. Nilkhamhang and N. Y. Chong. 'Learning social relations for culture aware interaction'. In: *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)* (2017).

[263] M. J. Pazzani and D. Billsus. 'Content-based recommendation systems'. In: *The adaptive web*. Springer, 2007, pages 325–341.

[264] M. Pecka and T. Svoboda. 'Safe exploration techniques for reinforcement learning–an overview'. In: *International Workshop on Modelling and Simulation for Autonomous Systems*. Springer. 2014, pages 357–375.

[265] F. Pedregosa et al. 'Scikit-learn: Machine Learning in Python'. In: *Journal of Machine Learning Research* 12 (2011), pages 2825–2830.

[266] A. Peine, A. Hallawa, J. Bickenbach, G. Dartmann, L. B. Fazlic, A. Schmeink, G. Ascheid, C. Thiemermann, A. Schuppert, R. Kindle et al. 'Development and validation of a reinforcement learning algorithm to dynamically optimize mechanical ventilation in critical care'. In: *NPJ digital medicine* 4.1 (2021), page 32.

[267] A. Pelc. 'Searching games with errors - fifty years of coping with liars'. In: *Theoretical Computer Science* 270.1-2 (2002), pages 71–109.

[268] B. Peng, Q. Jiao and T. Kurner. 'Angle of arrival estimation in dynamic indoor THz channels with Bayesian filter and reinforcement learning'. In: *2016 24th European Signal Processing Conference (EUSIPCO)* (2016).

[269] C. Peng and P. Vuorimaa. 'Automatic Navigation Among Mobile DTV Services'. In: *ICEIS*. 2004.

[270] C. Peng and P. Vuorimaa. 'Automatic Navigation Among Mobile DTV Services'. In: *ICEIS*. 2004.

[271] C. Perera, A. Zaslavsky, P. Christen and D. Georgakopoulos. 'Context aware computing for the internet of things: A survey'. In: *IEEE Communications Surveys & Tutorials* 16.1 (2014), pages 414–454.

[272] B. J. Pine, B. Victor and A. C. Boynton. 'Making mass customization work'. In: *Harvard business review* 71.5 (1993), pages 108–11.

[273] J. Pineau, M. G. Bellemare, A. J. Rush, A. Ghizaru and S. A. Murphy. 'Constructing evidence-based treatment strategies using methods from computer science'. In: *Drug and Alcohol Dependence* 88 (2007), S52–S60.

[274] J. Pineau, P. Vincent-Lamarre, K. Sinha, V. Larivière, A. Beygelzimer, F. d'Alché-Buc, E. Fox and H. Larochelle. 'Improving reproducibility in machine learning research (a report from the neurips 2019 reproducibility program)'. In: *The Journal of Machine Learning Research* 22.1 (2021), pages 7459–7478.

[275] A. Pnueli. 'The temporal logic of programs'. In: *18th Annual Symposium on Foundations of Computer Science*. IEEE, 1977, pages 46–57.

[276] A. Pnueli and R. Rosner. 'On the synthesis of a reactive module'. In: *ACM SIGPLAN-SIGACT*. 1989, pages 179–190.

[277] T. J. Pollard, A. E. Johnson, J. D. Raffa, L. A. Celi, R. G. Mark and O. Badawi. 'The eICU Collaborative Research Database, a freely available multi-center database for critical care research'. In: *Scientific data* 5.1 (2018), pages 1–13.

[278] A. Pomprapa, S. Leonhardt and B. J. Misgeld. 'Optimal learning control of oxygen saturation using a policy iteration algorithm and a proof-of-concept in an interconnecting three-tank system'. In: *Control Engineering Practice* 59 (2017), pages 194–203.

[279] S. Poria, E. Cambria, N. Howard, G.-B. Huang and A. Hussain. 'Fusing audio, visual and textual clues for sentiment analysis from multimodal content'. In: *Neurocomputing* 174 (2016), pages 50–59.

[280] N. Prasad, L.-F. Cheng, C. Chivers, M. Draugelis and B. E. Engelhardt. 'A Reinforcement Learning Approach to Weaning of Mechanical Ventilation in Intensive Care Units'. In: *CoRR* abs/1704.06300 (2017).

[281] N. Prasad, L. F. Cheng, C. Chivers, M. Draugelis and B. E. Engelhardt. 'A reinforcement learning approach to weaning of mechanical ventilation in intensive care units'. In: *33rd Conference on Uncertainty in Artificial Intelligence, UAI 2017*. 2017.

[282] M. Preda and D. Popescu. 'Personalized Web Recommendations: Supporting Epistemic Information about End-Users'. In: *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)* (2005).

[283] F. D. Priscoli, L. Fogliati, A. Palo and A. Pietrabissa. 'Dynamic Class of Service mapping for Quality of Experience control in future networks'. In: *WTC 2014; World Telecommunications Congress 2014*. VDE. 2014, pages 1–6.

[284] Z. Qin, I. Rishabh and J. Carnahan. 'A Scalable Approach for Periodical Personalized Recommendations'. In: *Proceedings of the 10th ACM Conference on Recommender Systems - RecSys '16* (2016).

[285] A. Raghu, O. Gottesman, Y. Liu, M. Komorowski, A. Faisal, F. Doshi-Velez and E. Brunskill. 'Behaviour Policy Estimation in Off-Policy Policy Evaluation: Calibration Matters'. In: *International Conference on Machine Learning (ICML) Workshop on CausalML* (2018).

[286] A. Raghu, M. Komorowski, L. A. Celi, P. Szolovits and M. Ghassemi. 'Continuous state-space models for optimal sepsis treatment: a deep reinforcement learning approach'. In: *Machine Learning for Healthcare Conference*. PMLR. 2017, pages 147–163.

[287] V. R. Raghuveer, B. K. Tripathy, T. Singh and S. Khanna. 'Reinforcement learning approach towards effective content recommendation in MOOC environments'. In: *2014 IEEE International Conference on MOOC, Innovation and Technology in Education (MITE)* (2014).

[288] P. P. Rao. 'A dynamic programming approach to determine optimal manpower recruitment policies'. In: *Journal of the Operational Research Society* 41.10 (1990), pages 983–988.

[289] E. Real, J. Shlens, S. Mazzocchi, X. Pan and V. Vanhoucke. 'YouTube-BoundingBoxes: A Large High-Precision Human-Annotated Data Set for Object Detection in Video'. In: (2017), pages 5296–5305.

[290] E. Rennison. 'Personalized Galaxies of Infomation'. In: *Companion of the ACM Conference on Human Factors in Computing Systems (CHI'95)*. 1995.

[291] P. Resnick and H. R. Varian. 'Recommender systems'. In: *Communications of the ACM* 40.3 (1997), pages 56–58.

[292] D. Riaño, M. Peleg and A. Ten Teije. 'Ten years of knowledge representation for health care (2009–2018): Topics, trends, and challenges'. In: *Artificial Intelligence in Medicine* 100 (2019), page 101713.

[293] F. Ricci, L. Rokach and B. Shapira. 'Introduction to recommender systems handbook'. In: *Recommender systems handbook*. Springer, 2011, pages 14–17.

[294] D. Riecken. 'Personalized views of personalization'. In: *Communications of the ACM* 43.8 (2000), pages 26–26.

[295] M. Riedmiller. 'Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method'. In: *European Conference on Machine Learning*. Springer. 2005, pages 317–328.

[296] H. Ritschel and E. André. 'Real-Time Robot Personality Adaptation based on Reinforcement Learning and Social Signals'. In: *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction - HRI '17* (2017).

[297] I. Rivas-Blanco, C. Lopez-Casado, C. J. Perez-del-Pulgar, F. Garcia-Vacas, J. C. Fraile and V. F. Munoz. 'Smart Cable-Driven Camera Robotic Assistant'. In: *IEEE Transactions on Human-Machine Systems* 48.2 (2018), pages 183–196.

[298] L. Roggeveen, A. El Hassouni, J. Ahrendt, T. Guo, L. Fleuren, P. Thoral, A. R. Girbes, M. Hoogendoorn and P. W. Elbers. 'Transatlantic transferability of a new reinforcement learning model for optimizing haemodynamic treatment for critically ill patients with sepsis'. In: *Artificial Intelligence in Medicine* 112 (2021), page 102003.

[299] D. M. Roijers, P. Vamplew, S. Whiteson and R. Dazeley. 'A survey of multi-objective sequential decision-making'. In: *Journal of Artificial Intelligence Research* 48 (2013), pages 67–113.

[300] P. Romer. *Human capital and growth: Theory and evidence*. 1989.

[301] N. Roy, J. Pineau and S. Thrun. 'Spoken dialogue management using probabilistic reasoning'. In: *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. 2000, pages 93–100.

[302] M. Rudary, S. Singh and M. E. Pollack. 'Adaptive cognitive orthotics'. In: *Twenty-first international conference on Machine learning - ICML '04* (2004).

[303] S. Russel. *Human Compatible: Artificial Intelligence and the Problem of Control*. Viking Penguin, 2019.

[304] S. Russell and P. Norvig. 'Artificial intelligence: a modern approach'. In: (2002).

[305] M. Ryan. 'Using abstract models of behaviours to automatically generate reinforcement learning hierarchies'. In: *ICML*. Volume 2. 2002, pages 522–529.

[306] D. Sadigh, E. Kim, S. Coogan, S. Sastry and S. Seshia. 'A learning based approach to control synthesis of markov decision processes for linear temporal logic specifications'. In: *CDC*. IEEE. 2014, pages 1091–1096.

[307] S. Saha and R. Quazi. 'Emotion-driven learning agent for setting rich presence in mobile telephony'. In: *2008 11th International Conference on Computer and Information Technology* (2008).

[308] R. Santa Cruz, F. Villarejo, C. Irrazabal and A. Ciapponi. 'High versus low positive end-expiratory pressure (PEEP) levels for mechanically ventilated adult patients with acute lung injury and acute respiratory distress syndrome'. In: *Cochrane Database of Systematic Reviews* 3 (2021).

[309] J. B. Schafer, D. Frankowski, J. Herlocker and S. Sen. 'Collaborative filtering recommender systems'. In: *The adaptive web*. Springer, 2007, pages 291–324.

[310] J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye and S. Young. 'Agenda-based user simulation for bootstrapping a POMDP dialogue system'. In: *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*. 2007, pages 149–152.

[311] T. Schaul, D. Horgan, K. Gregor and D. Silver. 'Universal value function approximators'. In: *International conference on machine learning*. PMLR. 2015, pages 1312–1320.

[312] A. Schmitt, B. Schatz and W. Minker. 'Modeling and Predicting Quality in Spoken Human-Computer Interaction'. In: *SIGDIAL Conference*. 2011.

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

[314] Y. A. Sekhavat. 'MPRL: Multiple-Periodic Reinforcement Learning for difficulty adjustment in rehabilitation games'. In: *2017 IEEE 5th International Conference on Serious Games and Applications for Health (SeGAH)* (2017).

[315] Y.-W. Seo and B.-T. Zhang. 'A reinforcement learning agent for personalized information filtering'. In: *Proceedings of the 5th international conference on Intelligent user interfaces*. ACM. 2000, pages 248–251.

[316] Y.-W. Seo and B.-T. Zhang. 'Learning user's preferences by analyzing Web-browsing behaviors'. In: *Proceedings of the fourth international conference on Autonomous agents*. ACM. 2000, pages 381–387.

[317] Y.-W. Seo and B.-T. Zhang. 'Learning user's preferences by analyzing Web-browsing behaviors'. In: *Proceedings of the fourth international conference on Autonomous agents*. ACM. 2000, pages 381–387.

[318] D. Shawky and A. Badawi. 'A Reinforcement Learning-Based Adaptive Learning System'. In: *Advances in Intelligent Systems and Computing* (2018), pages 221–231.

[319] S. Shen and M. Chi. 'Reinforcement Learning'. In: *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization - UMAP '16* (2016).

[320] H.-Y. Shum, X.-d. He and D. Li. 'From Eliza to XiaoIce: challenges and opportunities with social chatbots'. In: *Frontiers of Information Technology & Electronic Engineering* 19.1 (2018), pages 10–26.

[321] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel et al. 'A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play'. In: *Science* 362.6419 (2018), pages 1140–1144.

[322] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton et al. 'Mastering the game of go without human knowledge'. In: *nature* 550.7676 (2017), pages 354–359.

[323] D. Silver, S. Singh, D. Precup and R. S. Sutton. 'Reward is enough'. In: *Artificial Intelligence* 299 (2021), page 103535.

[324] C. Sing, P. Love and C. Tam. 'Stock-flow model for forecasting labor supply'. In: *Journal of Construction Engineering and Management* 138.6 (2012), pages 707–715.

[325] S. Singh. 'Transfer of learning by composing solutions of elemental sequential tasks'. In: *Machine Learning* 8.3 (1992), pages 323–339.

[326] M. Soleymani. 'Crowdsourcing for Affective Annotation of Video: Development of a Viewer-reported Boredom Corpus'. In: 2010.

[327] L. Song, W. Hsu, J. Xu and M. van der Schaar. 'Using Contextual Learning to Improve Diagnostic Accuracy: Application in Breast Cancer Screening'. In: *IEEE Journal of Biomedical and Health Informatics* 20.3 (2016), pages 902–914.

[328] N. Sprague and D. Ballard. 'Multiple-goal reinforcement learning with modular sarsa (0)'. In: (2003).

[329] A. R. Srinivasan and S. Chakraborty. 'Path planning with user route preference - A reward surface approximation approach using orthogonal Legendre polynomials'. In: *2016 IEEE International Conference on Automation Science and Engineering (CASE)* (2016).

[330] A. Srivihok and P. Sukonmanee. 'Intelligent Agent for e-Tourism: Personalization Travel Support Agent using Reinforcement Learning'. In: *WWW 2005*. 2005.

[331] A. Srivihok and P. Sukonmanee. 'Intelligent Agent for e-Tourism: Personalization Travel Support Agent using Reinforcement Learning'. In: *WWW 2005*. 2005.

[332] P.-H. Su, P. Budzianowski, S. Ultes, M. Gasic and S. Young. 'Sample-efficient Actor-Critic Reinforcement Learning with Supervised Data for Dialogue Management'. In: *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*. 2017, pages 147–157.

[333] P.-h. Su, Y.-B. Wang, T.-h. Yu and L.-s. Lee. 'A dialogue game framework with personalized training using reinforcement learning for computer-assisted language learning'. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (2013).

[334] P.-H. Su, C.-H. Wu and L.-S. Lee. 'A Recursive Dialogue Game for Personalized Computer-Aided Pronunciation Training'. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2014), pages 1–1.

[335] R. Sutton, D. Precup and S. Singh. 'Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning'. In: *Artificial Intelligence* 112.1-2 (1999), pages 181–211.

[336] R. S. Sutton. 'Generalization in reinforcement learning: Successful examples using sparse coarse coding'. In: *Advances in neural information processing systems*. 1996, pages 1038–1044.

[337] R. S. Sutton. 'Integrated architectures for learning, planning, and reacting based on approximating dynamic programming'. In: *Machine learning proceedings 1990*. Elsevier, 1990, pages 216–224.

[338] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[339] R. S. Sutton, D. A. McAllester, S. P. Singh and Y. Mansour. 'Policy gradient methods for reinforcement learning with function approximation'. In: *Advances in neural information processing systems*. 2000, pages 1057–1063.

[340] C. Szepesvári. 'Algorithms for reinforcement learning'. In: *Synthesis lectures on artificial intelligence and machine learning* 4.1 (2010), pages 1–103.

[341] S. A. Tabatabaei, M. Hoogendoorn and A. van Halteren. 'Narrowing Reinforcement Learning: Overcoming the Cold Start Problem for Personalized Health Interventions'. In: *International Conference on Principles and Practice of Multi-Agent Systems*. Springer. 2018, pages 312–327.

[342] N. Taghipour and A. Kardan. 'A hybrid web recommender system based on Q-learning'. In: *Proceedings of the 2008 ACM symposium on Applied computing - SAC '08* (2008).

[343] N. Taghipour, A. Kardan and S. S. Ghidary. 'Usage-based web recommendations'. In: *Proceedings of the 2007 ACM conference on Recommender systems - RecSys '07* (2007).

[344] L. Tang, Y. Jiang, L. Li and T. Li. 'Ensemble contextual bandits for personalized recommendation'. In: *Proceedings of the 8th ACM Conference on Recommender systems - RecSys '14* (2014).

[345] L. Tang, Y. Jiang, L. Li, C. Zeng and T. Li. 'Personalized Recommendation via Parameter-Free Contextual Bandits'. In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '15* (2015).

[346] L. Tang, R. Rosales, A. Singh and D. Agarwal. 'Automatic ad format selection via contextual bandits'. In: *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management - CIKM '13* (2013).

[347] S. Tang, A. Modi, M. Sjoding and J. Wiens. 'Clinician-in-the-loop decision making: Reinforcement learning with near-optimal set-valued policies'. In: *International Conference on Machine Learning*. PMLR. 2020, pages 9387–9396.

[348] M. Tavakol and U. Brefeld. 'A Unified Contextual Bandit Framework for Long- and Short-Term Recommendations'. In: *Lecture Notes in Computer Science* (2017), pages 269–284.

[349] B. Tegelund, H. Son and D. Lee. 'A task-oriented service personalization scheme for smart environments using reinforcement learning'. In: *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)* (2016).

[350] A. Ten Teije, S. Miksch and P. Lucas. *Computer-based medical guidelines and protocols: a primer and current trends*. Volume 139. Ios Press, 2008.

[351] G. Tesauro. 'TD-Gammon, a self-teaching backgammon program, achieves master-level play'. In: *Neural computation* 6.2 (1994), pages 215–219.

[352] G. Theocharous, P. S. Thomas and M. Ghavamzadeh. 'Personalized ad recommendation systems for life-time value optimization with guarantees'. In: *Twenty-Fourth International Joint Conference on Artificial Intelligence*. 2015.

[353] G. Theocharous, P. S. Thomas and M. Ghavamzadeh. 'Ad Recommendation Systems for Life-Time Value Optimization'. In: *Proceedings of the 24th International Conference on World Wide Web - WWW '15 Companion* (2015).

[354] P. Thomas and E. Brunskill. 'Data-efficient off-policy policy evaluation for reinforcement learning'. In: *International Conference on Machine Learning*. PMLR. 2016, pages 2139–2148.

[355] P. S. Thomas, G. Theocharous and M. Ghavamzadeh. 'High-confidence off-policy evaluation'. In: *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015.

[356] C. A. Thompson, M. H. Goker and P. Langley. 'A personalized system for conversational recommendations'. In: *Journal of Artificial Intelligence Research* 21 (2004), pages 393–428.

[357] P. J. Thoral et al. 'Sharing ICU Patient Data Responsibly Under the Society of Critical Care Medicine/European Society of Intensive Care Medicine Joint Data Science Collaboration: The Amsterdam University Medical Centers Database (AmsterdamUMCdb) Example'. In: *Critical care medicine* 49.6 (June 2021), e563–e577.

[358] S. Tomic, F. Pecora and A. Saffiotti. 'Learning Normative Behaviors through Abstraction'. In: *Proceedings of the 24th European Conference on Artificial Intelligence*. 2020.

[359] R. Toro Icarte, T. Klassen, R. Valenzano and S. McIlraith. 'Teaching multiple tasks to an RL agent using LTL'. In: *AAMAS*. 2018, pages 452–461.

[360] S. Triki and C. Hanachi. 'A Self-adaptive System for Improving Autonomy and Public Spaces Accessibility for Elderly'. In: *Smart Innovation, Systems and Technologies* (2017), pages 53–66.

[361] H.-H. Tseng, Y. Luo, S. Cui, J.-T. Chien, R. K. Ten Haken and I. E. Naqa. 'Deep reinforcement learning for automated radiation adaptation in lung cancer'. In: *Medical Physics* 44.12 (2017), pages 6690–6705.

[362] K. Tsiakas, C. Abellanoza and F. Makedon. 'Interactive Learning and Adaptation for Robot Assisted Therapy for People with Dementia'. In: *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments - PETRA '16* (2016).

[363] K. Tsiakas, M. Huber and F. Makedon. 'A multimodal adaptive session manager for physical rehabilitation exercising'. In: *Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments - PETRA '15* (2015).

[364] K. Tsiakas, M. Papakostas, B. Chebaa, D. Ebert, V. Karkaletsis and F. Makedon. 'An Interactive Learning and Adaptation Framework for Adaptive Robot Assisted Therapy'. In: *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments - PETRA '16* (2016).

[365] K. Tsiakas, M. Papakostas, M. Theofanidis, M. Bell, R. Mihalcea, S. Wang, M. Burzo and F. Makedon. 'An Interactive Multisensing Framework for Personalized Human Robot Collaboration and Assistive Training Using Reinforcement Learning'. In: *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments - PETRA '17* (2017).

[366] A. M. Turing. 'Computing Machinery and Intelligence'. In: *Mind* LIX.236 (Oct. 1950), pages 433–460.

[367] S. Ultes, A. Schmitt and W. Minker. 'On quality ratings for spoken dialogue systems–experts vs. users'. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2013, pages 569–578.

[368] D. Urieli and P. Stone. 'TacTex'13: a champion adaptive power trading agent'. In: *AAMAS*. 2014.

[369] W. M. Vagias. 'Likert-type scale response anchors. clemson international institute for tourism'. In: *& Research Development, Department of Parks, Recreation and Tourism Management, Clemson University* (2006).

[370] W. M. van der Aalst, M. Bichler and A. Heinzl. 'Responsible data science'. In: *Business & Information Systems Engineering* 59.5 (2017), pages 311–313.

[371] H. Van Hasselt, A. Guez and D. Silver. 'Deep reinforcement learning with double Q-learning'. In: *Thirtieth AAAI conference on artificial intelligence*. Volume 2. 2016, page 5.

[372] M. van Bekkum, M. de Boer, F. van Harmelen, A. Meyer-Vitali and A. t. Teije. 'Modular design patterns for hybrid learning and reasoning systems: a taxonomy, patterns and use cases'. In: *Applied Intelligence* 51.9 (2021), pages 6528–6546.

[373] G. Vasan and P. M. Pilarski. 'Learning from demonstration: Teaching a myoelectric prosthesis with an intact limb via reinforcement learning'. In: *2017 International Conference on Rehabilitation Robotics (ICORR)* (2017).

[374] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev et al. 'Grandmaster level in StarCraft II using multi-agent reinforcement learning'. In: *Nature* 575.7782 (2019), pages 350–354.

[375] P. Virtanen et al. 'SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python'. In: *Nature Methods* 17 (2020), pages 261–272.

[376] C. Voloshin, H. M. Le, N. Jiang and Y. Yue. 'Empirical Study of Off-Policy Policy Evaluation for Reinforcement Learning'. In: *Thirty-fifth Conference on Neural Information Processing Systems*. 2021.

[377] L. Wang, Y. Gao, C. Cao and L. Wang. 'Towards a General Supporting Framework for Self-Adaptive Software Systems'. In: *2012 IEEE 36th Annual Computer Software and Applications Conference Workshops* (2012).

[378] P. Wang, J. Rowe, B. Mott and J. Lester. 'Decomposing Drama Management in Educational Interactive Narrative: A Modular Reinforcement Learning Approach'. In: *Lecture Notes in Computer Science* (2016), pages 270–282.

[379] P. Wang, J. P. Rowe, W. Min, B. W. Mott and J. C. Lester. 'Interactive Narrative Personalization with Deep Reinforcement Learning'. In: *IJCAI*. 2017.

[380] X. Wang, Y. Wang, D. Hsu and Y. Wang. 'Exploration in Interactive Personalized Music Recommendation'. In: *ACM Transactions on Multimedia Computing, Communications, and Applications* 11.1 (2014), pages 1–22.

[381] X. Wang, Y. Wang, D. Hsu and Y. Wang. 'Exploration in interactive personalized music recommendation: a reinforcement learning approach'. In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 11.1 (2014), page 7.

[382] X. Wang, M. Zhang, F. Ren and T. Ito. 'GongBroker: A Broker Model for Power Trading in Smart Grid Markets'. In: *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)* (2015).

[383] C. J. Watkins. 'Learning from Delayed Rewards'. PhD thesis. King's College, 1989.

[384] C. J. Watkins and P. Dayan. 'Q-learning'. In: *Machine Learning* 8.3-4 (1992), pages 279–292.

[385] M. Wen, R. Ehlers and U. Topcu. 'Correct-by-synthesis reinforcement learning with temporal logic constraints'. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. RSJ/IEEE. 2015, pages 4983–4990.

[386] M. Wiering and M. Van Otterlo. 'Reinforcement learning'. In: *Adaptation, learning, and optimization* 12 (2012), page 3.

[387] J. Williams, A. Raux, D. Ramachandran and A. Black. 'The dialog state tracking challenge'. In: *Proceedings of the 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*. 2013, pages 404–413.

[388] J. D. Williams and S. Young. 'Partially observable Markov decision processes for spoken dialog systems'. In: *Computer Speech & Language* 21.2 (2007), pages 393–422.

[389] R. Williams. 'Simple statistical gradient-following algorithms for connectionist reinforcement learning'. In: *Machine Learning* 8.3 (1992), pages 229–256.

[390] G. Wu, Y. Ding, Y. Li, J. Luo, F. Zhang and J. Fu. 'Data-driven inverse learning of passenger preferences in urban public transits'. In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)* (2017).

[391] J. Wu, M. Li and C.-H. Lee. 'An entropy minimization framework for goal-driven dialogue management'. In: *Sixteenth Annual Conference of the International Speech Communication Association*. 2015.

[392] P. R. Wurman, S. Barrett, K. Kawamoto, J. MacGlashan, K. Subramanian, T. J. Walsh, R. Capobianco, A. Devlic, F. Eckert, F. Fuchs et al. 'Outracing champion Gran Turismo drivers with deep reinforcement learning'. In: *Nature* 602.7896 (2022), pages 223–228.

[393] J. Xu, T. Xing and M. van der Schaar. 'Personalized Course Sequence Recommendations'. In: *IEEE Transactions on Signal Processing* 64.20 (2016), pages 5340–5352.

[394] F. Yang, D. Lyu, B. Liu and S. Gustafson. 'PEORL: integrating symbolic planning and hierarchical reinforcement learning for robust decision-making'. In: *IJCAI*. 2018, pages 4860–4866.

[395] M. Yang, Q. Qu, K. Lei, J. Zhu, Z. Zhao, X. Chen and J. Z. Huang. 'Investigating Deep Reinforcement Learning Techniques in Personalized Dialogue Generation'. In: *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM. 2018, pages 630–638.

[396] M. Yang, W. Tu, Q. Qu, Z. Zhao, X. Chen and J. Zhu. 'Personalized response generation by Dual-learning based domain adaptation'. In: *Neural Networks* 103 (2018), pages 72–82.

[397] M. Yang, Z. Zhao, W. Zhao, X. Chen, J. Zhu, L. Zhou and Z. Cao. 'Personalized Response Generation via Domain adaptation'. In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '17* (2017).

[398] W.-C. Yang, G. Marra, G. Rens and L. De Raedt. 'Safe Reinforcement Learning via Probabilistic Logic Shields'. In: (Aug. 2023). Edited by E. Elkind. Main Track, pages 5739–5749.

[399] Z. Yang, B. Li, Y. Zhu, I. King, G. Levow and H. Meng. 'Collection of user judgments on spoken dialog system with crowdsourcing'. In: *2010 IEEE Spoken Language Technology Workshop*. IEEE. 2010, pages 277–282.

[400] S. Young, M. Gašić, B. Thomson and J. D. Williams. 'POMDP-based statistical spoken dialog systems: A review'. In: *Proceedings of the IEEE* 101.5 (2013), pages 1160–1179.

[401] C. Yu, J. Liu and H. Zhao. 'Inverse reinforcement learning for intelligent mechanical ventilation and sedative dosing in intensive care units'. In: *BMC medical informatics and decision making* 19.2 (2019), pages 111–120.

[402] S.-T. Yuan. 'A personalized and integrative comparison-shopping engine and its applications'. In: *Decision Support Systems* 34.2 (2003), pages 139–156.

[403] Y. Yue, S. A. Hong and C. Guestrin. 'Hierarchical exploration for accelerating contextual bandits'. In: *Proceedings of the 29th International Coference on International Conference on Machine Learning*. Omnipress. 2012, pages 979–986.

[404] S. Zaidenberg and P. Reignier. 'Reinforcement Learning of User Preferences for a Ubiquitous Personal Assistant'. In: *Advances in Reinforcement Learning*. IntechOpen, 2011.

[405] S. Zaidenberg, P. Reignier and J. L. Crowley. 'Reinforcement Learning of Context Models for a Ubiquitous Personal Assistant'. In: *3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008* (2008), pages 254–264.

[406] A. Zapf, S. Castell, L. Morawietz and A. Karch. 'Measuring inter-rater reliability for nominal data–which coefficients and confidence intervals are appropriate?' In: *BMC medical research methodology* 16.1 (2016), page 93.

[407] C. Zeng, Q. Wang, S. Mokhtari and T. Li. 'Online Context-Aware Recommendation with Time Varying Multi-Armed Bandit'. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16* (2016).

[408] B.-T. Zhang and Y.-W. Seo. 'Personalized web-document filtering using reinforcement learning'. In: *Applied Artificial Intelligence* 15.7 (2001), pages 665–685.

[409] B.-T. Zhang and Y.-W. Seo. 'Personalized web-document filtering using reinforcement learning'. In: *Applied Artificial Intelligence* 15.7 (2001), pages 665–685.

[410] H. Zhang, Z. Gao, Y. Zhou, H. Zhang, K. Wu and F. Lin. 'Faster and Safer Training by Embedding High-Level Knowledge into Deep Reinforcement Learning'. In: *arXiv preprint arXiv:1910.09986* (2019).

[411] Y. Zhang, R. Chen, J. Tang, W. F. Stewart and J. Sun. 'LEAP'. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '17* (2017).

[412] T. Zhao and I. King. 'Locality-Sensitive Linear Bandit Model for Online Social Recommendation'. In: *Lecture Notes in Computer Science* (2016), pages 80–90.

[413] Y. Zhao, Q. Zhao, L. Xia, Z. Cheng, F. Wang and F. Song. 'A unified control framework of HVAC system for thermal and acoustic comforts in office building'. In: *2013 IEEE International Conference on Automation Science and Engineering (CASE)* (2013).

[414] Y. Zhao, S. Wang, Y. Zou, J. Ng and T. Ng. 'Automatically Learning User Preferences for Personalized Service Composition'. In: *2017 IEEE International Conference on Web Services (ICWS)* (2017).

[415] Y. Zhao, M. R. Kosorok and D. Zeng. 'Reinforcement learning design for cancer clinical trials'. In: *Statistics in medicine* 28.26 (2009), pages 3294–3315.

[416] Y. Zhao, D. Zeng, M. A. Socinski and M. R. Kosorok. 'Reinforcement Learning Strategies for Clinical Trials in Nonsmall Cell Lung Cancer'. In: *Biometrics* 67.4 (2011), pages 1422–1433.

[417] G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie and Z. Li. 'DRN'. In: *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18* (2018).

[418] H. Zheng and J. Jumadinova. 'OWLS: Observational Wireless Life-enhancing System (Extended Abstract)'. In: *AAMAS*. 2016.

[419] L. Zhou and E. Brunskill. 'Latent Contextual Bandits and their Application to Personalized Recommendations for New Users'. In: *IJCAI*. 2016.

[420] M. Zhou, Y. D. Mintz, Y. Fukuoka, K. Y. Goldberg, E. Flowers, P. Kaminsky, A. Castillejo and A. Aswani. 'Personalizing Mobile Fitness Apps using Reinforcement Learning'. In: *IUI Workshops*. 2018.

[421] R. Zhu, Y.-Q. Zhao, G. Chen, S. Ma and H. Zhao. 'Greedy outcome weighted tree learning of optimal personalized treatment rules'. In: *Biometrics* 73.2 (2016), pages 391–400.

B

Online References

[422] European Commission. *Directive 2014/65/EU of the European Parliament and of the Council of 15 May 2014 on markets in financial instruments and amending Directive 2002/92/EC and Directive 2011/61/EU (recast) Text with EEA relevance*. Retrieved 2023-02-02. 2014. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32014L0065.

[423] European Commission. *On Artificial Intelligence-A European Approach to Excellence and Trust*. Retrieved on 2022-10-03. 2020. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52020DC0065.

[424] M. Kennedy. *Computer Learns To Play Go At Superhuman Levels 'Without Human Knowledge'*. Retrieved on 2023-02-02. 2017. URL: https://www.npr.org/sections/thetwo-way/2017/10/18/558519095/computer-learns-to-play-go-at-superhuman-levels-without-human-knowledge.

[425] OpenAI. *ChatGPT: Optimizing Language Models for Dialogue*. Retrieved on 2022-12-23. 2023. URL: https://openai.com/blog/chatgpt/.

[426] J. Stolze et al. *https://app.ai-cursus.nl/home*. Retrieved on 2022-10-03. 2022. URL: https://app.ai-cursus.nl/home.

B

Summary

Artificial Intelligence (AI) studies how intelligence found in nature can be mimicked in machines. It does not really matter what type of machine is used, but computers have proven to be very flexible and useful for this purpose. In nature, intelligent agents are remarkable for their ability to change their behavior based on past experiences. It is a good idea to make machines do the same. This is the core idea behind the field of *machine learning*. Machine learning knows many different techniques and set-ups, but one of the most generic and powerful forms of machine learning is *reinforcement learning* (RL). RL is a very general form of machine learning because it learns how to take a sequence of actions based on a sequence of situations. The actions chosen should be the best according to a score that is assigned to actions over time.

RL has recently attracted a lot of attention, both inside and outside of the academic world. Many of you will recognize the 'AI' that has beat world champions in the games of Go, Chess and Starcraft-II. RL has also been used to improve computer chip designs, a very challenging job in which human experts have developed deep expertise over decades. Additionally, RL has been used to predict how protein structures behave and to minimize the energy used by data centers. These real-world successes are impressive and inspiring. They show how RL can improve our lives. However, RL has mostly seen real-world successes in highly controlled settings that involve humans only to a very limited extent. This is why in this thesis we examine the usage of RL in human contexts.

We do so by first looking at some applications of RL in human contexts to better understand *why* it may be challenging to use RL in human contexts in Part I. We then use the learned lessons to improve RL. We improve RL by combining it with existing knowledge encoded in symbols that are human-readable in Part II. Let us dive into these parts in more detail next.

Applications of Reinforcement Learning in Human Contexts

If you want to improve something, it is a good idea to first look at what has already been done. We do so with a look at the academic literature on the usage of RL to adapt digital systems to individuals in Chapter 2. We review existing work in a systematic way to make sure that we do not miss any important papers, so that everyone can understand how the discussed work was selected and so that it becomes clear how existing work is similar or complementary to each other.

A first proposal for applying RL in human contexts can be found in Chapter 3, where we study how we can personalize how chatbots decide what to say next. We personalize a chatbot that gives advice on financial products. We compare two personalization alternatives that both use RL in a different way. The first alternative groups users and then trains a chatbot per group, whereas the second alternative trains a single chatbot and blends chat histories with characteristics of the user to achieve personalization. We found that a personalized approach works well in the financial domain and that personalized chatbots outperform fixed decision rules in our evaluation, which were previously considered to be the best decision-making approach. In this study we used a simulator so that we could easily compare approaches.

A human behavior simulator may not be available for all chatbots that could benefit from personalization. If such a simulator is absent, scoring example dialogues of the chatbot has to be done manually, which is what the designers of the recently popular line of Chat-GPT bots have done. Manual scoring can be an expensive and tedious task. This is why it is important that the manual scoring process is designed well. This is why we looked into the collection of high-quality dialogue user satisfaction ratings in Chapter 4. We collect best practices for obtaining these labels from literature, compare definitions of 'user satisfaction', and test two different user interfaces for scoring dialogues. We find that our interfaces provide high quality dialogue scores. We shared the source code for this tool under a permissive license, so that anyone interested in collecting high quality dialogue user ratings can use it.

A second proposal to apply RL in human contexts is to use RL to select which kinds of jobs to open in Chapter 5. Organizations need to have the right people in the right place at the right time to meet their goals. This is challenging for three reasons. First, making the right decisions now requires taking the future into account. If an organization has many highly experienced employees that will retire soon, they should hire some employees with reasonable experience and train them further on the job. Second, many aspects of the organization are hard to predict: Who knows when employees will want to leave their job? Third, it is hard to explicitly specify what a good workforce looks like. To address these challenges, we combine deep neural networks with RL. A first benefit of our approach is that it allows HR specialists to define goals in high-level terms that they are familiar with, such as the ratio of managers to nonmanagers. The second key benefit is that the approach performs well

when employees quit or move jobs unexpectedly, as we found in an evaluation on real-world data.

While we were working on the puzzles described above, we noticed that some important pieces to make an impact with RL in human contexts are still missing. We study these pieces in more depth in part 2.

Subsymbolic RL and Symbolic Knowledge

The theoretical and algorithmic advances that we propose in this work combine techniques in two different directions within AI. These directions have often been presented as conflicting in the past, even though these directions have roots in antiquity and even both contain technologies that have around much longer than the term 'AI' itself. This has changed in recent times within a new field of research that looks at the combination of these so-called *symbolic* and *subsymbolic* directions. Before diving into the specific contributions presented in this thesis, we will briefly look into these directions and see why it is so promising to combine them.

Symbolic AI refers to approaches that aim to create machines with intelligence based on high-level symbolic (human-readable) representations of problems and solutions. For example, a human-readable representation of the concept 'cat' can consist of its name, whether it is neutered and its relation with other concepts such as the humans that feed it, the address it lives at, and its preferred toy. The field of symbolic AI has produced techniques with many strengths. Many of these techniques, for example, break a big problem down into ever-smaller problems until the final problems are so small that the solution is clear immediately. Doing so gives us an understanding of how the technique is tackling the problem and allows us to inspect its reasoning at any point in time. Additionally, symbolic AI approaches typically handle the arrival of new (symbolic) knowledge very well and can often adapt an existing solution to a new situation immediately. However, weaknesses of symbolic AI are that it can be difficult to describe situations and appropriate behavior with upfront and that it has shown insufficient performance on associative tasks with little inherent structure such as perception and locomotion.

Subsymbolic AI approaches also aim to create intelligence in machines. However, the representations used in these techniques are not human-readable. As you will understand, it is quite hard to give an example of a representation of a cat that is not human readable. Let me try anyway, using our current understanding of how the brains of humans and cats operate. When we observe a cat we know, the neurons in our brain exhibit a particular activation pattern that somehow produces the sensation of recognition. This particular pattern is a representation of a cat that seems to work to us individually, but that we cannot share with others easily or manipulate explicitly, because it does consist of symbols. Subsymbolic AI can learn similarly uninterpretable representations based on how well these work rather than how well we can understand them. The strength of subsymbolic AI is that it performs very well on associative tasks such as perception and locomotion, but that it is much harder to employ

on highly structured and high-level tasks such formal reasoning and planning, tasks where prior knowledge is available, and tasks where outcomes have to be correct to ensure the safety of people, animals and artificially intelligent agents.

Since subsymbolic and symbolic AI techniques offer complementary strengths, it is a good idea to combine them. This is what we do in Chapter 6. Here, we address the problem of choosing the settings of mechanical ventilators at the intensive care unit (ICU). Mechanical ventilators are devices that supply air with oxygen to and remove air with carbon dioxide from the lungs of patients for whom their spontaneous breathing is inadequate to maintain life. These devices can be configured in various ways, for example, to blow more (or less) air into the lungs at a lower (or higher) pressure and containing more (or less) oxygen. By learning good settings from those previously selected, we may be able to reduce costs and improve quality of care. However, we already have an understanding of good strategies for mechanical ventilation. For example, we know which settings *never* to use. We therefore encode treatment advice from a medical guideline using logic from symbolic AI and then supply it to an RL algorithm that relies on subsymbolic representations. We make the solution safe by limiting the settings for the agent to choose from, and we change the reward signal based on descriptions in the guideline. We evaluated the solutions obtained using real-world data that were collected previously. The results show that our solution chooses settings that are more varied than settings chosen by clinicians, and more safe. Although the evaluation of previously collected data proved challenging, the results indicate that the solution found in this way would increase the survival rate of patients receiving mechanical ventilation in the ICU.

Having looked at prior knowledge about safety and RL, we turn back to our earlier application of the chatbot for recommending financial products. In this application, we also want to control our agent and ensure that it protects customers' best interests. Similarly to the ICU example, knowledge about product advice to customers has been encoded into a guideline that any representative of the bank has to adhere to. Since a chatbot advisor can be seen as a representative of a bank, it also has to follow the guideline. A key difference between the medical and the financial guidelines is that the medical guideline describes only (un)desirable behaviors per situation, whereas the financial one also describes (un)desirable behaviors over time. An example of a restriction that includes time is: "*always* verify the identity of the customers *before* you give advice". This is why, in Chapter 7, we looked at making sure that we can learn safely with RL if safety includes the notion of time. We showed how these restrictions can harm the ability to learn suitable behavior, and proposed an algorithm that bypasses negative effects of constraints with symbolic AI. We showed that this algorithm performs almost equally as an unsafe algorithm, but that it does not cause safety violations.

While it is important to ensure that RL agents adhere to *restrictive* instructions as we have done so far, it is also interesting to look at affirmative instructions. In Chapter 8 we develop a framework for instructing RL agents. The instructions we focus on here are in some way similar to recipes: they

describe high-level steps that you need to perform to reach the goal of a spectacular dish, but leave specifics out. As a result, it is possible to follow all steps to the letter but still end up with a miserable dish. We, again used symbolic AI to represent and reason over the instructions and combined it with a subsymbolic technique called *deep RL*. However, these techniques were combined in a different way from our earlier approaches. Instead of altering the reward function and limiting the actions available to the agent, we split the subsymbolic solution up into smaller components based on the symbolic knowledge. Instead of learning a single solution for "preparing zucchini soup", we learn single solutions for its sub-parts "chop zucchini", "pour into pan", "blend until smooth" and of course "season to taste". These separate solutions can also be reused across tasks and reconfigured to solve previously unseen tasks: the agent does not need to start from scratch and can get to suitable solutions more quickly as a result. It should be noted here that there are no predefined connections between the subtask names such as "chop zucchini" or "mix" and the associated behaviors. The agent learns these associations autonomously when given a sufficiently rich set of tasks – it has in some sense successfully learned what it means to "chop zucchini" by itself by breaking up the full task into smaller pieces with symbolic high-level reasoning and then learned solutions for these smaller pieces with subsymbolic RL.

Samenvatting

Artificiële Intelligentie (AI) bestudeert hoe intelligentie in de natuur kan worden nagebootst in machines. Voor 'machines' kan je ook 'computers' lezen, want die zijn lekker flexibel en worden daarom veel gebruikt. In de natuur vallen intelligente wezens op doordat ze hun gedrag kunnen aanpassen aan de hand van opgedane ervaringen. Daarom is het een goed idee om aan machines ook een aanpassingsvermogen mee te geven. Dit is het kern-idee van het studieveld van het *machinaal leren*. Er zijn verschillende vorm van machinaal leren, waarvan *reinforcement learning* (RL) er eentje is. RL is een erg algemene vorm van machinaal leren omdat het draait om het opeenvolgend kiezen van een actie voor een situatie. De acties moeten zo goed mogelijk zijn, waarbij met een score aangegeven wordt wat goede acties waren.

Er is recentelijk veel aandacht voor RL, zowel in als buiten de academische wereld. Velen van jullie zullen de 'AI' herkennen die wereldkampioenen versloeg in de spelletjes go, schaak en het computerspel 'Starcraft-II'. RL is ook gebruikt om ontwerpen van computer chips te verbeteren, een uitdagende taak waar menselijke experts al decennia lang op puzzelen. RL is ook gebruikt om te voorspellen hoe eiwitstructuren zich gedragen en om het energieverbruik van datacentra omlaag te krijgen. Deze toepassingen zijn inspirirend, want ze laten zien hoe RL ons leven kan verbeteren. Echter, veel praktijksuccesses van RL zijn behaald in sterk gecontroleerde omgevingen waar mensen maar beperkt bij komen kijken. Daarom onderzoeken we in dit proefschrift het gebruik van RL in menselijke omgevingen.

We kijken eerst naar toepassingen van RL in menselijke omgevingen zodat we beter begrijpen *waarom* het zo lastig is gebleken om RL in menselijke omgevingen toe te passen in Deel I. Daarna gebruiken we de opgedane lessen om RL te verbeteren in Deel II. Dat doen we door het te combineren met voorkennis in door mensen leesbare, symbolische beschrijvingen. Hieronder vind je een beschrijving van deze twee delen.

Toepassingen van Reinforcement Learning in Menselijke Omgevingen

Als je iets wil verbeteren, is het een goed idee om eerst te kijken wat anderen allemaal al geprobeerd hebben. Dat doen we in dit proefschrift door te kijken naar wetenschappelijke literatuur over het gebruik van RL om digitale systemen aan te passen aan individuen in Hoofdstuk 2. We geven een systematisch overzicht van eerder werk zodat we geen belangrijke publicaties missen, zodat iedereen snapt hoe het overzicht tot stand is gekomen en zodat het duidelijk wordt welke technieken op elkaar lijken en welke elkaar juist aanvullen.

Een eerste voorstel om RL toe te passen in een menselijke context is te vinden in Hoofdstuk 3, waarin we praatprogramma's[1] personalizeren. We doen dit aan de hand van een praatprogramma die advies kan geven over financiële producten. We vergelijken twee manieren die ieder op een eigen manier RL gebruiken. De eerste manier verdeelt gebruikers in groepen aan de hand van hun kenmerken en leert dan een programma per groep. De tweede manier traint één programma getraind en voegt gebruikerskenmerken toe aan de geschiedenis van het gesprek. Als we de programma's testen, zien we dat ze het goed doen in het financiële domein en dat gepersonaliseerde versies het beter doen dan programma's met de voorheen beste oplossing, welke beslissingen maakt op basis van vooraf vastgelegde regels. In deze studie gebruikten we een simulator zodat we de verschillende programma's goed konden vergelijken.

Helaas is er niet voor ieder praatprogramma dat gepersonaliseerd kan worden een simulator. Als er geen simulator is, kan het praatprogramma getraind worden op scores die door mensen zijn gegeven– dit is wat de ontwerpers van het recentelijk populaire praatprogramma chat-GPT hebben gedaan. Het geven van zulke scores is duur en kost veel tijd. Daarom is het belangrijk dat dit proces goed in elkaar steekt. We kijken daarom naar het verzamelen van tevredenheidsscores van praatprogramma-gebruikers in Hoofdstuk 4. We verzamelen praktijk-adviezen voor het scoren van gesprekken uit wetenschappelijke literatuur, vergelijken verschillende tevredenheidsscores en testen twee nieuwe gebruikersomgevingen[2] voor het geven van scores. Beide leveren scores van hoge kwaliteit op. We delen alle broncode zodat iedereen die dat wil bruikbare tevredenheidsscores kan verzamelen.

Een tweede voorstel voor het toepassen van RL in een menselijke omgeving is te vinden in Hoofdstuk 5, waar we bestuderen of RL gebruikt kan worden om te bepalen wat er voor vacatures geopend moeten worden. Om hun doelen te behalen moeten organisaties de juiste mensen op de juiste plek zien te krijgen op het juiste moment. Dit is lastig vanwege drie factoren. Allereerst moet er vooruit gekeken worden. Als er bijvoorbeeld veel ervaren medewerkers zijn die binnenkort met pensioen zullen gaan, dan is het een goed idee om nu alvast aan vervanging te gaan denken. Ten tweede is vooruit kijken lastig, omdat de loopbaan van medewerkers zich vaak lastig laat voorspellen. Tot slot is het ook nog lastig om aan te geven hoe een geschikt personeelsbestand er precies uit ziet.

[1] chatbots
[2] user interfaces

Om al deze uitdagingen te lijf te gaan, gebruiken we een combinatie van RL met grote neurale netwerken[3]. Hiermee kunnen HR specialisten hun doelen stellen aan de hand van kengetallen waar ze bekend mee zijn, zoals bijvoorbeeld de hoeveelheid leidinggevenden t.o.v. de hoeveelheid niet-leidinggevenden. Daar bovenop doet onze oplossing het goed wanneer medewerkers onverwacht van baan wisselen of ontslag nemen, zo blijkt uit een praktijkstudie.

Door aan bovenstaande puzzels te werken, kregen we inzicht over missende stukjes voor impact met RL in menselijke omgevingen. We bestuderen deze stukjes van dichtbij in het volgende deel.

Subsymbolische RL en Symbolische Kennis

De theoretische en algoritmische bijdragen die we in dit deel van het werk presenteren combineren twee verschillende richtingen binnen AI. Hoewel beide richtingen uit de oudheid stammen en technieken bevatten die veel langer bestaan de term 'AI', zijn deze richtingen in het verleden vaak als tegenstrijdig gepresenteerd. Dit is de laatste tijd aan het veranderen met een nieuw onderzoeksgebied dat zich bezighoudt met de combinatie van deze zogenaamde *symbolische* en *subsymbolische* richtingen. Voor we verder duiken in de specifieke bijdragen in dit proefschrift, kijken we kort naar deze twee richtingen waarbij we zullen zien waarom het veelbelovend is ze te combineren.

Symbolische AI houdt zich bezig met het maken van intelligente machines aan de hand van abstracte symbolische beschrijvingen van problemen en oplossingen, zodat wij mensen die kunnen begrijpen. Een voorbeeld van een symbolische beschrijving van een kat bestaat uit diens naam, of de kat gecastreerd is en de relatie tot andere concepten zoals de mensen die het beestje eten geven, het adres of het favoriete speeltje. Binnen de symbolische AI zijn technieken met vele voordelen ontwikkeld, zoals bijvoorbeeld de mogelijkheid om inzicht te geven in de manier waarop een probleem wordt opgelost door deze op te breken in kleine brokjes met kleine deeloplossingen. Hiernaast zijn veel symbolische AI technieken erg sterk in het verwerken van nieuwe (symbolische) kennis: ze kunnen dan vaak direct tot een geschikte oplossing komen. Nadelen, daarintegen, zijn dat het vaak lastig is om alle situaties en geschikte acties op voorhand te beschrijven. Ook zijn deze technieken niet erg geschikt gebleken voor associatieve en ongestructureerde taken zoals waarneming en beweging.

Subsymbolische AI houdt zich ook bezig met het maken van intelligente machines, maar gebruikt daarvoor beschrijvingen die niet voor mensen begrijpelijk hoeven te zijn. Zoals je je kunt voorstellen, is het lastig om een voorbeeld te geven van een beschrijving van een kat die niet door mensen te begrijpen is. Laat het me toch proberen, aan de hand van ons begrip van de werking van hersenen van mensen en katten. Als we een bekende kat zien, dan ontstaat er in de neuronen van ons brein een activitatiepatroon dat op één of andere manier bij ons de sensatie van herkenning oproept. Dit specifieke patroon is een beschrijving van de kat die voor ons goed werkt maar die we niet met

[3] rekenmodellen geinspireerd op het brein

anderen kunnen delen of expliciet kunnen manipuleren omdat deze beschrijving niet uit symbolen bestaat. Subsymbolische AI doet iets soortgelijks en kiest ook beschrijvingen die goed werken maar niet per sé voor begrijpelijk zijn voor mensen. Een voordeel hiervan is dat het erg goed werkt op ongestructureerde en associatieve taken zoals waarneming en beweging, maar het is helaas wel erg lastig toe te passen op gestructureerde en abstracte taken zoals logisch redeneren en plannen, taken waar al kennis over beschikbaar is en taken waar uitkomsten correct *moeten* zijn vanwege de veiligheid van mens, dier en artificiële wezens.

Aangezien subsymbolische en symbolische AI technieken elkaar aanvullen, is het een goed idee om ze te combineren. Dat is dan ook wat we doen in Hoofdstuk 6. In dit hoofdstuk werken we aan het probleem van het kiezen van instellingen voor beademingsapparaten op de intensive care-afdeling (IC) van het ziekenhuis. Beademingsapparaten leveren zuurstofrijke lucht aan de longen van patiënten van wie het eigen vermogen om te ademen te beperkt is om te leven, en ze zuigen lucht met koolstofdioxide weer weg. Als we kunnen leren wat goede instellingen zijn voor verschillende patiënten, dan kunnen we zorgkosten naar beneden brengen en de zorg verbeteren. We hoeven hier echter niet vanaf het begin te beginnen. We hebben namelijk al een behoorlijk begrip van (on)geschikte instellingen van beademingsapparaten in medische richtlijnen. We gebruiken daarom een symbolische beschrijving van deze richtlijnen om een subsymbolische RL oplossing te verbeteren: we maken deze veilig door de RL oplossing te laten kiezen uit veilige instellingen en we baseren het leer-signaal op symbolische beschrijvingen uit de richtlijn. We evalueren de gevonden oplossingen met eerder verzamelde patiëntgegevens en zien dat de gevonden oplossing instellingen kiest die veiliger en gevarieerder zijn dan de instellingen die clinici op de IC meestal doen. Hoewel een evaluatie op eerder verzamelde gegevens uitdagend is, laten de resultaten wel zien dat de gevonden oplossing voor de overlevingskansen van longpatiënten op de IC zou kunnen verhogen.

S

Nu we een voorbeeld van veilig leren met RL hebben gezien, grijpen we terug op de eerdere toepassing van het praatprogramma voor financiële producten. Van dit praatprogramma willen namelijk ook garanderen dat het bij het geven van advies de belangen van de klant behartigt. Net zoals bij de beademing, gebruiken we hiervoor een bestaande richtlijn die beschrijft hoe vertegenwoordigers van een bank zich moeten gedragen tegenover klanten. Het praatprogramma is ook een vertegenwoordiger, dus je zou kunnen zeggen dat die zich ook aan deze richtlijn zou moeten houden. Een belangrijk verschil tussen de medische richtlijn en de financiële is dat de medische richtlijn beschrijft of acties wenselijk zijn per situatie, terwijl de financiële richtlijn ook beschrijft of acties wenselijk aan de hand van eerdere situaties en acties. Bijvoorbeeld: "controleer *altijd* de identiteid van de klant *voordat* je een product aanbeveelt." Daarom zorgen we er in Hoofdstuk 7 voor dat we met RL veilig kunnen leren als de veiligheidseisen eerdere acties en situaties mee nemen. We laten in dit hoofdstuk zien hoe zulke veiligheidseisen het leervermogen kunnen schaden en stellen een algoritme voor om dit te omzeilen aan de hand van symbolische AI.

We laten zien dat dit veilige algoritme het bijna net zo goed doet als onveilige algoritmes, maar dan zonder veiligheidsblunders te maken.

Alhoewel het nuttig is om ervoor te zorgen dat RL zich kan houden aan *limiterende* instructies, kan het ook interessant zijn om te kijken naar *bevestigende* instructioes. In Hoofdstuk 8 ontwikkelen we een manier om zulke instructies aan RL te geven. Hierbij nemen we instructies die lijken op recepten: ze beschrijven wel de stappen om een fantastisch gerecht te maken, maar laten de details aan de kok. Het is daarom mogelijk om alle stappen netjes te volgens, maar toch met een jammerlijk gerecht te eindigen. We gebruiken ook hier weer symbolische AI voor de instructies en combineren het met RL. We doen dat deze keer door met symbolische RL de taak van het koken van bijvoorbeeld courgette-soep op te splitsen in kleinere taken zoals het snijden van courgette, die we vervolgens aan de hand van RL oplossen. De kleinere oplossingen kunnen hergebruikt worden om bijvoorbeeld geheel nieuwe taken op te lossen, bijvoorbeeld het snijden van courgette voor een salade. Daar komt bij dat de oplossingen voor kleinere taken automatisch worden geleerd als de gegeven instructies gevarieerd genoeg zijn – je zou kunnen zeggen dat er geleerd wordt wat het betekent als er bijvoorbeeld 'snijd de courgette in blokjes' in een recept staat door deze combinatie van symbolisch redeneren en subsymbolische RL.

SIKS Dissertatiereeks

2016 01 Syed Saiden Abbas (RUN), Recognition of Shapes by Humans and Machines
02 Michiel Christiaan Meulendijk (UU), Optimizing medication reviews through decision support: prescribing a better pill to swallow
03 Maya Sappelli (RUN), Knowledge Work in Context: User Centered Knowledge Worker Support
04 Laurens Rietveld (VU), Publishing and Consuming Linked Data
05 Evgeny Sherkhonov (UVA), Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers
06 Michel Wilson (TUD), Robust scheduling in an uncertain environment
07 Jeroen de Man (VU), Measuring and modeling negative emotions for virtual training
08 Matje van de Camp (TiU), A Link to the Past: Constructing Historical Social Networks from Unstructured Data
09 Archana Nottamkandath (VU), Trusting Crowdsourced Information on Cultural Artefacts
10 George Karafotias (VUA), Parameter Control for Evolutionary Algorithms
11 Anne Schuth (UVA), Search Engines that Learn from Their Users
12 Max Knobbout (UU), Logics for Modelling and Verifying Normative Multi-Agent Systems
13 Nana Baah Gyan (VU), The Web, Speech Technologies and Rural Development in West Africa - An ICT4D Approach
14 Ravi Khadka (UU), Revisiting Legacy Software System Modernization
15 Steffen Michels (RUN), Hybrid Probabilistic Logics - Theoretical Aspects, Algorithms and Experiments
16 Guangliang Li (UVA), Socially Intelligent Autonomous Agents that Learn from Human Reward

17 Berend Weel (VU), Towards Embodied Evolution of Robot Organisms
18 Albert Meroño Peñuela (VU), Refining Statistical Data on the Web
19 Julia Efremova (Tu/e), Mining Social Structures from Genealogical Data
20 Daan Odijk (UVA), Context & Semantics in News & Web Search
21 Alejandro Moreno Célleri (UT), From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground
22 Grace Lewis (VU), Software Architecture Strategies for Cyber-Foraging Systems
23 Fei Cai (UVA), Query Auto Completion in Information Retrieval
24 Brend Wanders (UT), Repurposing and Probabilistic Integration of Data; An Iterative and data model independent approach
25 Julia Kiseleva (TU/e), Using Contextual Information to Understand Searching and Browsing Behavior
26 Dilhan Thilakarathne (VU), In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains
27 Wen Li (TUD), Understanding Geo-spatial Information on Social Media
28 Mingxin Zhang (TUD), Large-scale Agent-based Social Simulation - A study on epidemic prediction and control
29 Nicolas Höning (TUD), Peak reduction in decentralised electricity systems - Markets and prices for flexible planning
30 Ruud Mattheij (UvT), The Eyes Have It
31 Mohammad Khelghati (UT), Deep web content monitoring
32 Eelco Vriezekolk (UT), Assessing Telecommunication Service Availability Risks for Crisis Organisations
33 Peter Bloem (UVA), Single Sample Statistics, exercises in learning from just one example
34 Dennis Schunselaar (TUE), Configurable Process Trees: Elicitation, Analysis, and Enactment
35 Zhaochun Ren (UVA), Monitoring Social Media: Summarization, Classification and Recommendation
36 Daphne Karreman (UT), Beyond R2D2: The design of nonverbal interaction behavior optimized for robot-specific morphologies
37 Giovanni Sileno (UvA), Aligning Law and Action - a conceptual and computational inquiry
38 Andrea Minuto (UT), Materials that Matter - Smart Materials meet Art & Interaction Design
39 Merijn Bruijnes (UT), Believable Suspect Agents; Response and Interpersonal Style Selection for an Artificial Suspect
40 Christian Detweiler (TUD), Accounting for Values in Design

41 Thomas King (TUD), Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance
42 Spyros Martzoukos (UVA), Combinatorial and Compositional Aspects of Bilingual Aligned Corpora
43 Saskia Koldijk (RUN), Context-Aware Support for Stress Self-Management: From Theory to Practice
44 Thibault Sellam (UVA), Automatic Assistants for Database Exploration
45 Bram van de Laar (UT), Experiencing Brain-Computer Interface Control
46 Jorge Gallego Perez (UT), Robots to Make you Happy
47 Christina Weber (UL), Real-time foresight - Preparedness for dynamic innovation networks
48 Tanja Buttler (TUD), Collecting Lessons Learned
49 Gleb Polevoy (TUD), Participation and Interaction in Projects. A Game-Theoretic Analysis
50 Yan Wang (UVT), The Bridge of Dreams: Towards a Method for Operational Performance Alignment in IT-enabled Service Supply Chains

---

2017 01 Jan-Jaap Oerlemans (UL), Investigating Cybercrime
02 Sjoerd Timmer (UU), Designing and Understanding Forensic Bayesian Networks using Argumentation
03 Daniël Harold Telgen (UU), Grid Manufacturing; A Cyber-Physical Approach with Autonomous Products and Reconfigurable Manufacturing Machines
04 Mrunal Gawade (CWI), Multi-core Parallelism in a Column-store
05 Mahdieh Shadi (UVA), Collaboration Behavior
06 Damir Vandic (EUR), Intelligent Information Systems for Web Product Search
07 Roel Bertens (UU), Insight in Information: from Abstract to Anomaly
08 Rob Konijn (VU) , Detecting Interesting Differences:Data Mining in Health Insurance Data using Outlier Detection and Subgroup Discovery
09 Dong Nguyen (UT), Text as Social and Cultural Data: A Computational Perspective on Variation in Text
10 Robby van Delden (UT), (Steering) Interactive Play Behavior
11 Florian Kunneman (RUN), Modelling patterns of time and emotion in Twitter #anticipointment
12 Sander Leemans (TUE), Robust Process Mining with Guarantees
13 Gijs Huisman (UT), Social Touch Technology - Extending the reach of social touch through haptic technology
14 Shoshannah Tekofsky (UvT), You Are Who You Play You Are: Modelling Player Traits from Video Game Behavior
15 Peter Berck (RUN), Memory-Based Text Correction

16 Aleksandr Chuklin (UVA), Understanding and Modeling Users of Modern Search Engines
17 Daniel Dimov (UL), Crowdsourced Online Dispute Resolution
18 Ridho Reinanda (UVA), Entity Associations for Search
19 Jeroen Vuurens (UT), Proximity of Terms, Texts and Semantic Vectors in Information Retrieval
20 Mohammadbashir Sedighi (TUD), Fostering Engagement in Knowledge Sharing: The Role of Perceived Benefits, Costs and Visibility
21 Jeroen Linssen (UT), Meta Matters in Interactive Storytelling and Serious Gaming (A Play on Worlds)
22 Sara Magliacane (VU), Logics for causal inference under uncertainty
23 David Graus (UVA), Entities of Interest — Discovery in Digital Traces
24 Chang Wang (TUD), Use of Affordances for Efficient Robot Learning
25 Veruska Zamborlini (VU), Knowledge Representation for Clinical Guidelines, with applications to Multimorbidity Analysis and Literature Search
26 Merel Jung (UT), Socially intelligent robots that understand and respond to human touch
27 Michiel Joosse (UT), Investigating Positioning and Gaze Behaviors of Social Robots: People's Preferences, Perceptions and Behaviors
28 John Klein (VU), Architecture Practices for Complex Contexts
29 Adel Alhuraibi (UvT), From IT-BusinessStrategic Alignment to Performance: A Moderated Mediation Model of Social Innovation, and Enterprise Governance of IT"
30 Wilma Latuny (UvT), The Power of Facial Expressions
31 Ben Ruijl (UL), Advances in computational methods for QFT calculations
32 Thaer Samar (RUN), Access to and Retrievability of Content in Web Archives
33 Brigit van Loggem (OU), Towards a Design Rationale for Software Documentation: A Model of Computer-Mediated Activity
34 Maren Scheffel (OU), The Evaluation Framework for Learning Analytics
35 Martine de Vos (VU), Interpreting natural science spreadsheets
36 Yuanhao Guo (UL), Shape Analysis for Phenotype Characterisation from High-throughput Imaging
37 Alejandro Montes Garcia (TUE), WiBAF: A Within Browser Adaptation Framework that Enables Control over Privacy
38 Alex Kayal (TUD), Normative Social Applications
39 Sara Ahmadi (RUN), Exploiting properties of the human auditory system and compressive sensing methods to increase noise robustness in ASR
40 Altaf Hussain Abro (VUA), Steer your Mind: Computational Exploration of Human Control in Relation to Emotions, Desires and Social Support For applications in human-aware support systems

	41	Adnan Manzoor (VUA), Minding a Healthy Lifestyle: An Exploration of Mental Processes and a Smart Environment to Provide Support for a Healthy Lifestyle
	42	Elena Sokolova (RUN), Causal discovery from mixed and missing data with applications on ADHD datasets
	43	Maaike de Boer (RUN), Semantic Mapping in Video Retrieval
	44	Garm Lucassen (UU), Understanding User Stories - Computational Linguistics in Agile Requirements Engineering
	45	Bas Testerink (UU), Decentralized Runtime Norm Enforcement
	46	Jan Schneider (OU), Sensor-based Learning Support
	47	Jie Yang (TUD), Crowd Knowledge Creation Acceleration
	48	Angel Suarez (OU), Collaborative inquiry-based learning
2018	01	Han van der Aa (VUA), Comparing and Aligning Process Representations
	02	Felix Mannhardt (TUE), Multi-perspective Process Mining
	03	Steven Bosems (UT), Causal Models For Well-Being: Knowledge Modeling, Model-Driven Development of Context-Aware Applications, and Behavior Prediction
	04	Jordan Janeiro (TUD), Flexible Coordination Support for Diagnosis Teams in Data-Centric Engineering Tasks
	05	Hugo Huurdeman (UVA), Supporting the Complex Dynamics of the Information Seeking Process
	06	Dan Ionita (UT), Model-Driven Information Security Risk Assessment of Socio-Technical Systems
	07	Jieting Luo (UU), A formal account of opportunism in multi-agent systems
	08	Rick Smetsers (RUN), Advances in Model Learning for Software Systems
	09	Xu Xie (TUD), Data Assimilation in Discrete Event Simulations
	10	Julienka Mollee (VUA), Moving forward: supporting physical activity behavior change through intelligent technology
	11	Mahdi Sargolzaei (UVA), Enabling Framework for Service-oriented Collaborative Networks
	12	Xixi Lu (TUE), Using behavioral context in process mining
	13	Seyed Amin Tabatabaei (VUA), Computing a Sustainable Future
	14	Bart Joosten (UVT), Detecting Social Signals with Spatiotemporal Gabor Filters
	15	Naser Davarzani (UM), Biomarker discovery in heart failure
	16	Jaebok Kim (UT), Automatic recognition of engagement and emotion in a group of children
	17	Jianpeng Zhang (TUE), On Graph Sample Clustering
	18	Henriette Nakad (UL), De Notaris en Private Rechtspraak
	19	Minh Duc Pham (VUA), Emergent relational schemas for RDF
	20	Manxia Liu (RUN), Time and Bayesian Networks

	21	Aad Slootmaker (OUN), EMERGO: a generic platform for authoring and playing scenario-based serious games
	22	Eric Fernandes de Mello Araújo (VUA), Contagious: Modeling the Spread of Behaviours, Perceptions and Emotions in Social Networks
	23	Kim Schouten (EUR), Semantics-driven Aspect-Based Sentiment Analysis
	24	Jered Vroon (UT), Responsive Social Positioning Behaviour for Semi-Autonomous Telepresence Robots
	25	Riste Gligorov (VUA), Serious Games in Audio-Visual Collections
	26	Roelof Anne Jelle de Vries (UT),Theory-Based and Tailor-Made: Motivational Messages for Behavior Change Technology
	27	Maikel Leemans (TUE), Hierarchical Process Mining for Scalable Software Analysis
	28	Christian Willemse (UT), Social Touch Technologies: How they feel and how they make you feel
	29	Yu Gu (UVT), Emotion Recognition from Mandarin Speech
	30	Wouter Beek, The "K" in "semantic web" stands for "knowledge": scaling semantics to the web
2019	01	Rob van Eijk (UL),Web privacy measurement in real-time bidding systems. A graph-based approach to RTB system classification
	02	Emmanuelle Beauxis Aussalet (CWI, UU), Statistics and Visualizations for Assessing Class Size Uncertainty
	03	Eduardo Gonzalez Lopez de Murillas (TUE), Process Mining on Databases: Extracting Event Data from Real Life Data Sources
	04	Ridho Rahmadi (RUN), Finding stable causal structures from clinical data
	05	Sebastiaan van Zelst (TUE), Process Mining with Streaming Data
	06	Chris Dijkshoorn (VU), Nichesourcing for Improving Access to Linked Cultural Heritage Datasets
	07	Soude Fazeli (TUD), Recommender Systems in Social Learning Platforms
	08	Frits de Nijs (TUD), Resource-constrained Multi-agent Markov Decision Processes
	09	Fahimeh Alizadeh Moghaddam (UVA), Self-adaptation for energy efficiency in software systems
	10	Qing Chuan Ye (EUR), Multi-objective Optimization Methods for Allocation and Prediction
	11	Yue Zhao (TUD), Learning Analytics Technology to Understand Learner Behavioral Engagement in MOOCs
	12	Jacqueline Heinerman (VU), Better Together
	13	Guanliang Chen (TUD), MOOC Analytics: Learner Modeling and Content Generation
	14	Daniel Davis (TUD), Large-Scale Learning Analytics: Modeling Learner Behavior & Improving Learning Outcomes in Massive Open Online Courses

15 Erwin Walraven (TUD), Planning under Uncertainty in Constrained and Partially Observable Environments
16 Guangming Li (TUE), Process Mining based on Object-Centric Behavioral Constraint (OCBC) Models
17 Ali Hurriyetoglu (RUN),Extracting actionable information from microtexts
18 Gerard Wagenaar (UU), Artefacts in Agile Team Communication
19 Vincent Koeman (TUD), Tools for Developing Cognitive Agents
20 Chide Groenouwe (UU), Fostering technically augmented human collective intelligence
21 Cong Liu (TUE), Software Data Analytics: Architectural Model Discovery and Design Pattern Detection
22 Martin van den Berg (VU),Improving IT Decisions with Enterprise Architecture
23 Qin Liu (TUD), Intelligent Control Systems: Learning, Interpreting, Verification
24 Anca Dumitrache (VU), Truth in Disagreement - Crowdsourcing Labeled Data for Natural Language Processing
25 Emiel van Miltenburg (VU), Pragmatic factors in (automatic) image description
26 Prince Singh (UT), An Integration Platform for Synchromodal Transport
27 Alessandra Antonaci (OUN), The Gamification Design Process applied to (Massive) Open Online Courses
28 Esther Kuindersma (UL), Cleared for take-off: Game-based learning to prepare airline pilots for critical situations
29 Daniel Formolo (VU), Using virtual agents for simulation and training of social skills in safety-critical circumstances
30 Vahid Yazdanpanah (UT), Multiagent Industrial Symbiosis Systems
31 Milan Jelisavcic (VU), Alive and Kicking: Baby Steps in Robotics
32 Chiara Sironi (UM), Monte-Carlo Tree Search for Artificial General Intelligence in Games
33 Anil Yaman (TUE), Evolution of Biologically Inspired Learning in Artificial Neural Networks
34 Negar Ahmadi (TUE), EEG Microstate and Functional Brain Network Features for Classification of Epilepsy and PNES
35 Lisa Facey-Shaw (OUN), Gamification with digital badges in learning programming
36 Kevin Ackermans (OUN), Designing Video-Enhanced Rubrics to Master Complex Skills
37 Jian Fang (TUD), Database Acceleration on FPGAs
38 Akos Kadar (OUN), Learning visually grounded and multilingual representations

---

2020 01 Armon Toubman (UL), Calculated Moves: Generating Air Combat Behaviour

02 Marcos de Paula Bueno (UL), Unraveling Temporal Processes using Probabilistic Graphical Models
03 Mostafa Deghani (UvA), Learning with Imperfect Supervision for Language Understanding
04 Maarten van Gompel (RUN), Context as Linguistic Bridges
05 Yulong Pei (TUE), On local and global structure mining
06 Preethu Rose Anish (UT), Stimulation Architectural Thinking during Requirements Elicitation - An Approach and Tool Support
07 Wim van der Vegt (OUN), Towards a software architecture for reusable game components
08 Ali Mirsoleimani (UL),Structured Parallel Programming for Monte Carlo Tree Search
09 Myriam Traub (UU), Measuring Tool Bias and Improving Data Quality for Digital Humanities Research
10 Alifah Syamsiyah (TUE), In-database Preprocessing for Process Mining
11 Sepideh Mesbah (TUD), Semantic-Enhanced Training Data AugmentationMethods for Long-Tail Entity Recognition Models
12 Ward van Breda (VU), Predictive Modeling in E-Mental Health: Exploring Applicability in Personalised Depression Treatment
13 Marco Virgolin (CWI), Design and Application of Gene-pool Optimal Mixing Evolutionary Algorithms for Genetic Programming
14 Mark Raasveldt (CWI/UL), Integrating Analytics with Relational Databases
15 Konstantinos Georgiadis (OUN), Smart CAT: Machine Learning for Configurable Assessments in Serious Games
16 Ilona Wilmont (RUN), Cognitive Aspects of Conceptual Modelling
17 Daniele Di Mitri (OUN), The Multimodal Tutor: Adaptive Feedback from Multimodal Experiences
18 Georgios Methenitis (TUD), Agent Interactions & Mechanisms in Markets with Uncertainties: Electricity Markets in Renewable Energy Systems
19 Guido van Capelleveen (UT), Industrial Symbiosis Recommender Systems
20 Albert Hankel (VU), Embedding Green ICT Maturity in Organisations
21 Karine da Silva Miras de Araujo (VU), Where is the robot?: Life as it could be
22 Maryam Masoud Khamis (RUN), Understanding complex systems implementation through a modeling approach: the case of e-government in Zanzibar
23 Rianne Conijn (UT), The Keys to Writing: A writing analytics approach to studying writing processes using keystroke logging
24 Lenin da Nóbrega Medeiros (VUA/RUN), How are you feeling, human? Towards emotionally supportive chatbots
25 Xin Du (TUE), The Uncertainty in Exceptional Model Mining

	26	Krzysztof Leszek Sadowski (UU), GAMBIT: Genetic Algorithm for Model-Based mixed-Integer opTimization
	27	Ekaterina Muravyeva (TUD), Personal data and informed consent in an educational context
	28	Bibeg Limbu (TUD), Multimodal interaction for deliberate practice: Training complex skills with augmented reality
	29	Ioan Gabriel Bucur (RUN), Being Bayesian about Causal Inference
	30	Bob Zadok Blok (UL), Creatief, Creatiever, Creatiefst
	31	Gongjin Lan (VU), Learning better – From Baby to Better
	32	Jason Rhuggenaath (TUE), Revenue management in online markets: pricing and online advertising
	33	Rick Gilsing (TUE), Supporting service-dominant business model evaluation in the context of business model innovation
	34	Anna Bon (MU), Intervention or Collaboration? Redesigning Information and Communication Technologies for Development
	35	Siamak Farshidi (UU), Multi-Criteria Decision-Making in Software Production
2021	01	Francisco Xavier Dos Santos Fonseca (TUD),Location-based Games for Social Interaction in Public Space
	02	Rijk Mercuur (TUD), Simulating Human Routines: Integrating Social Practice Theory in Agent-Based Models
	03	Seyyed Hadi Hashemi (UVA), Modeling Users Interacting with Smart Devices
	04	Ioana Jivet (OU), The Dashboard That Loved Me: Designing adaptive learning analytics for self-regulated learning
	05	Davide Dell'Anna (UU), Data-Driven Supervision of Autonomous Systems
	06	Daniel Davison (UT), "Hey robot, what do you think?" How children learn with a social robot
	07	Armel Lefebvre (UU), Research data management for open science
	08	Nardie Fanchamps (OU), The Influence of Sense-Reason-Act Programming on Computational Thinking
	09	Cristina Zaga (UT), The Design of Robothings. Non-Anthropomorphic and Non-Verbal Robots to Promote Children's Collaboration Through Play
	10	Quinten Meertens (UvA), Misclassification Bias in Statistical Learning
	11	Anne van Rossum (UL), Nonparametric Bayesian Methods in Robotic Vision
	12	Lei Pi (UL), External Knowledge Absorption in Chinese SMEs
	13	Bob R. Schadenberg (UT), Robots for Autistic Children: Understanding and Facilitating Predictability for Engagement in Learning
	14	Negin Samaeemofrad (UL), Business Incubators: The Impact of Their Support

15 Onat Ege Adali (TU/e), Transformation of Value Propositions into Resource Re-Configurations through the Business Services Paradigm
16 Esam A. H. Ghaleb (UM), Bimodal emotion recognition from audio-visual cues
17 Dario Dotti (UM), Human Behavior Understanding from motion and bodily cues using deep neural networks
18 Remi Wieten (UU), Bridging the Gap Between Informal Sense-Making Tools and Formal Systems - Facilitating the Construction of Bayesian Networks and Argumentation Frameworks
19 Roberto Verdecchia (VU), Architectural Technical Debt: Identification and Management
20 Masoud Mansoury (TU/e), Understanding and Mitigating Multi-Sided Exposure Bias in Recommender Systems
21 Pedro Thiago Timbó Holanda (CWI), Progressive Indexes
22 Sihang Qiu (TUD), Conversational Crowdsourcing
23 Hugo Manuel Proença (LIACS), Robust rules for prediction and description
24 Kaijie Zhu (TUE), On Efficient Temporal Subgraph Query Processing
25 Eoin Martino Grua (VUA), The Future of E-Health is Mobile: Combining AI and Self-Adaptation to Create Adaptive E-Health Mobile Applications
26 Benno Kruit (CWI & VUA), Reading the Grid: Extending Knowledge Bases from Human-readable Tables
27 Jelte van Waterschoot (UT), Personalized and Personal Conversations: Designing Agents Who Want to Connect With You
28 Christoph Selig (UL), Understanding the Heterogeneity of Corporate Entrepreneurship Programs

---

2022 01 Judith van Stegeren (UT), Flavor text generation for role-playing video games
02 Paulo da Costa (TU/e), Data-driven Prognostics and Logistics Optimisation: A Deep Learning Journey
03 Ali el Hassouni (VUA), A Model A Day Keeps The Doctor Away: Reinforcement Learning For Personalized Healthcare
04 Ünal Aksu (UU), A Cross-Organizational Process Mining Framework
05 Shiwei Liu (TU/e), Sparse Neural Network Training with In-Time Over-Parameterization
06 Reza Refaei Afshar (TU/e), Machine Learning for Ad Publishers in Real Time Bidding
07 Sambit Praharaj (OU), Measuring the Unmeasurable? Towards Automatic Co-located Collaboration Analytics
08 Maikel L. van Eck (TU/e), Process Mining for Smart Product Design
09 Oana Andreea Inel (VUA), Understanding Events: A Diversity-driven Human-Machine Approach

10 Felipe Moraes Gomes (TUD), Examining the Effectiveness of Collaborative Search Engines
11 Mirjam de Haas (UT), Staying engaged in child-robot interaction, a quantitative approach to studying preschoolers' engagement with robots and tasks during second-language tutoring
12 Guanyi Chen (UU), Computational Generation of Chinese Noun Phrases
13 Xander Wilcke (VUA), Machine Learning on Multimodal Knowledge Graphs: Opportunities, Challenges, and Methods for Learning on Real-World Heterogeneous and Spatially-Oriented Knowledge
14 Michiel Overeem (UU), Evolution of Low-Code Platforms
15 Jelmer Jan Koorn (UU), Work in Process: Unearthing Meaning using Process Mining
16 Pieter Gijsbers (TU/e), Systems for AutoML Research
17 Laura van der Lubbe (VUA), Empowering vulnerable people with serious games and gamification
18 Paris Mavromoustakos Blom (TiU), Player Affect Modelling and Video Game Personalisation
19 Bilge Yigit Ozkan (UU), Cybersecurity Maturity Assessment and Standardisation
20 Fakhra Jabeen (VUA), Dark Side of the Digital Media - Computational Analysis of Negative Human Behaviors on Social Media
21 Seethu Mariyam Christopher (UM), Intelligent Toys for Physical and Cognitive Assessments
22 Alexandra Sierra Rativa (TiU), Virtual Character Design and its potential to foster Empathy, Immersion, and Collaboration Skills in Video Games and Virtual Reality Simulations
23 Ilir Kola (TUD), Enabling Social Situation Awareness in Support Agents
24 Samaneh Heidari (UU), Agents with Social Norms and Values - A framework for agent based social simulations with social norms and personal values
25 Anna L.D. Latour (LU), Optimal decision-making under constraints and uncertainty
26 Anne Dirkson (LU), Knowledge Discovery from Patient Forums: Gaining novel medical insights from patient experiences
27 Christos Athanasiadis (UM), Emotion-aware cross-modal domain adaptation in video sequences
28 Onuralp Ulusoy (UU), Privacy in Collaborative Systems
29 Jan Kolkmeier (UT), From Head Transform to Mind Transplant: Social Interactions in Mixed Reality
30 Dean De Leo (CWI), Analysis of Dynamic Graphs on Sparse Arrays
31 Konstantinos Traganos (TU/e), Tackling Complexity in Smart Manufacturing with Advanced Manufacturing Process Management
32 Cezara Pastrav (UU), Social simulation for socio-ecological systems

	33	Brinn Hekkelman (CWI/TUD), Fair Mechanisms for Smart Grid Congestion Management
	34	Nimat Ullah (VUA), Mind Your Behaviour: Computational Modelling of Emotion & Desire Regulation for Behaviour Change
	35	Mike E.U. Ligthart (VUA), Shaping the Child-Robot Relationship: Interaction Design Patterns for a Sustainable Interaction
2023	01	Bojan Simoski (VUA), Untangling the Puzzle of Digital Health Interventions
	02	Mariana Rachel Dias da Silva (TiU), Grounded or in flight? What our bodies can tell us about the whereabouts of our thoughts
	03	Shabnam Najafian (TUD), User Modeling for Privacy-preserving Explanations in Group Recommendations
	04	Gineke Wiggers (UL), The Relevance of Impact: bibliometric-enhanced legal information retrieval
	05	Anton Bouter (CWI), Optimal Mixing Evolutionary Algorithms for Large-Scale Real-Valued Optimization, Including Real-World Medical Applications
	06	António Pereira Barata (UL), Reliable and Fair Machine Learning for Risk Assessment
	07	Tianjin Huang (TU/e), The Roles of Adversarial Examples on Trustworthiness of Deep Learning
	08	Lu Yin (TU/e), Knowledge Elicitation using Psychometric Learning
	09	Xu Wang (VUA), Scientific Dataset Recommendation with Semantic Techniques
	10	Dennis J.N.J. Soemers (UM), Learning State-Action Features for General Game Playing
	11	Fawad Taj (VUA), Towards Motivating Machines: Computational Modeling of the Mechanism of Actions for Effective Digital Health Behavior Change Applications
	12	Tessel Bogaard (VUA), Using Metadata to Understand Search Behavior in Digital Libraries
	13	Injy Sarhan (UU), Open Information Extraction for Knowledge Representation
	14	Selma Čaušević (TUD), Energy resilience through self-organization
	15	Alvaro Henrique Chaim Correia (TU/e), Insights on Learning Tractable Probabilistic Graphical Models
	16	Peter Blomsma (TiU), Building Embodied Conversational Agents: Observations on human nonverbal behaviour as a resource for the development of artificial characters
	17	Meike Nauta (UT), Explainable AI and Interpretable Computer Vision – From Oversight to Insight
	18	Gustavo Penha (TUD), Designing and Diagnosing Models for Conversational Search and Recommendation
	19	George Aalbers (TiU), Digital Traces of the Mind: Using Smartphones to Capture Signals of Well-Being in Individuals

20 Arkadiy Dushatskiy (TUD), Expensive Optimization with Model-Based Evolutionary Algorithms applied to Medical Image Segmentation using Deep Learning
21 Gerrit Jan de Bruin (UL), Network Analysis Methods for Smart Inspection in the Transport Domain
22 Alireza Shojaifar (UU), Volitional Cybersecurity
23 Theo Theunissen (UU), Documentation in Continuous Software Development
24 Agathe Balayn (TUD), Practices Towards Hazardous Failure Diagnosis in Machine Learning
25 Jurian Baas (UU), Entity Resolution on Historical Knowledge Graphs
26 Loek Tonnaer (TU/e), Linearly Symmetry-Based Disentangled Representations and their Out-of-Distribution Behaviour
27 Ghada Sokar (TU/e), Learning Continually Under Changing Data Distributions
28 Floris den Hengst (VUA), Learning to Behave: Reinforcement Learning in Human Contexts