universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Optimizing Exposure Therapy Protocols Using Reinforcement Learning

verfasst von | submitted by

Lisa Tomasiak B.Sc.

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2024

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 910
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Computational Science
Betreut von | Supervisor:	Assoz. Prof. Dipl.-Ing. Dr.techn. Sebastian Tschiatschek BSc
Mitbetreut von | Co-Supervisor:	Dr. Filip Melinščak BSc MSc

Acknowledgements

Throughout my years of education I have received great support from various sources. First, I want to thank my parents for their financial but especially for their emotional support over my years of study. I am beyond grateful for their continuous assistance and trust in my efforts. I also want to express deep gratitude to my partner for his continuous and unconditional encouragement and his comprehensive support and faith in me. Furthermore, I am grateful for numerous activities with friends lifting me up and keeping my overall motivation high. Lastly, I want to express my appreciation to my supervisors Ass.-Prof. Dr. Sebastian Tschiatschek and Dr. Filip Melinscak, BSc MSc for the ongoing discussions and their content-related contributions to this thesis.

Danksagung

Während meiner Studienjahre habe ich von unterschiedlichen Quellen große Unterstützung erfahren. Der erste Dank gilt meinen Eltern für ihre finanzielle, aber vor allem für ihre emotionale Unterstützung über den gesamten Zeitraum meiner Ausbildung hinweg. Ich bin zutiefst dankbar für ihren anhaltenden Beistand und ihr Vertrauen in meine Bemühungen. Mein tiefer Dank gilt auch meinem Partner für seinen kontinuierlichen und bedingungslosen Zuspruch, seine umfassende Unterstützung und seinen anhaltenden Glauben an mich. Ich bin weiters dankbar für die vielen gemeinsamen Unternehmungen mit meinem Freundeskreis, der laufend für Aufmunterung gesorgt und meine allgemeine Motivation hoch gehalten hat. Letztlich möchte ich meine Wertschätzung gegenüber meinen Betreuern Ass.-Prof. Dr. Sebastian Tschiatschek und Dr. Filip Melinscak, BSc MSc ausdrücken, die durch ihre inhaltlichen Beiträge und unsere anhaltenden Diskussionen wesentlich zu dieser Masterthesis beigetragen haben.

Abstract

Although Reinforcement Learning (RL) has shown promise in healthcare [76, 79, 141], its potential in improving mental health treatments has not yet been explored comprehensively. The aim of this master thesis is to investigate the potential of RL algorithms for the optimization of Exposure Therapy (ET) protocols, which are commonly used for treating Anxiety Disorders and Anxiety-Related Disorders, such as phobias. Specifically, simulation studies are conducted where computational associative learning models (such as the Rescorla-Wagner model [115]) are used as a stand-in for patients and an RL algorithm aims to optimize an ET protocol with the goal of robustly extinguishing negative associations. To this end, an investigation on different patient model parameters, RL environmental settings and RL model hyperparameters is performed. The results suggest that parameter settings like the number of extinction trials and the formulation of the reward function have a significant impact on the RL agent's ability to find an efficient ET protocol successfully extinguishing a patient's fear and that optimized simulation settings can be transferred from one patient model to other related ones, achieving comparable results.

Kurzfassung

Während sich Reinforcement Learning (RL) bereits erfolgreich für einige Anwendungen im Gesundheitswesen erwiesen hat [76, 79, 141], ist dessen Potential im Bereich der mentalen Gesundheit noch nicht umfassend erforscht. Ziel dieser Masterarbeit ist die Untersuchung der Nutzbarkeit von RL Algorithmen für die Optimierung von Protokollen für Expositionstherapien, der Standardbehandlung bei Angststörungen wie z.B. Phobien. In der vorliegenden Arbeit werden Simulationen durchgeführt, in denen PatientInnen durch mathematische, assoziative Lernmodelle (z.B. das Rescorla-Wagner Modell [115]) repräsentiert werden und ein RL Algorithmus versucht Expositionstherapie-Protokolle zu optimieren, mit dem Ziel negative Assoziationen wirksam auszulöschen. Untersuchungen verschiedener Parameter der Patientenmodelle, Einstellungen bezüglich der RL Umgebung, sowie Hyperparameter des RL Modells werden präsentiert. Die Ergebnisse zeigen, dass einzelne Parameter, wie die Anzahl der Expositionen pro Expositionsprotokoll oder die Formulierung der Reward-Funktion, einen bedeutenden Einfluss auf die Fähigkeit des RL Agenten haben, erfolgreich ein Angst reduzierendes Expositionsprotokoll zu identifizieren. Es konnte weiters gezeigt werden, dass Simulations-Einstellungen, die für ein Patientenmodell optimiert wurden, auf andere, ähnliche Patientenmodelle übertragbar sind und dabei zu vergleichbaren Ergebnissen führen.

Contents

List of Tables

List of Figures

List of Algorithms

1 Introduction

In recent years, the integration of Artificial Intelligence (AI) into the healthcare sector has changed the way we approach diagnostics, treatment, and therapy [59, 67]. A general aim of AI techniques in medicine is to uncover clinically relevant information from health care data and to assist clinical decision making [95, 97], with the potential of reducing diagnostic and therapeutic errors [40, 102]. Furthermore, AI assisted healthcare can effectively reduce the cost and risk of medical procedures, improve the utilization efficiency of medical resources and make medical services ubiquitous [41]. Instead of deriving treatment regimes from the average population response, the usage of AI in healthcare offers the opportunity to find precise treatment for individual patients with a variety in disease severity, personal characteristics and drug sensitivity [19, 40, 97, 102]. Promising synergy effects have emerged between AI techniques and precision medicine, where AI has supported tasks like disease diagnosis, and predicting risk and treatment response [68].

While AI technology is becoming more prevalent in physical health applications [101], the discipline of mental health has been slower in utilizing AI [54, 67, 91]. A possible reason is the fact that mental health practitioners are relying more on skills like forming relationships with patients and directly observing patient behaviors and emotions [46]. However, the future of AI in mental healthcare is promising [123], as it can help to redefine diagnosis and to better understand mental illnesses [26].

The most prevalent mental disorders are Anxiety Disorders and Anxiety-Related Disorders (ADs), which are associated with a high burden of illness [64, 74]. The primary therapy recommended for addressing ADs is cognitive-behavioral therapy, specifically Exposure Therapy (ET), which involves gradually and safely exposing individuals to the source of their anxiety [24]. Even though it is frequently considered the most effective therapeutic option, approximately half of the patients either fail to respond to ET or encounter lingering symptoms and relapses [80, 126].

There has been a general effort by researchers to improve ET by focusing on Extinction Learning (EL) models, which involve the experimental induction and subsequent extinction of negative associations, aligning with Pavlovian principles of fear acquisition and ET respectively [36, 103]. Research investigating the optimization of ET protocols demonstrated encouraging outcomes [118], yet subsequent replication studies and follow-up analyses have produced more ambiguous results [27, 77]. One possible reason for the difficulties in reproducibility is the unsystematic variability of procedural variables between studies due to uncertainty about optimal conditions and a lack of formalized specifications for ET protocols. While there are general design guidelines for extinction protocols [81], many questions regarding procedural variables in ET, such as the optimal duration and spacing of sessions, the efficacy of graded versus random intensities of

exposure stimuli, and whether stimuli are best presented individually or in combinations, remain without conclusive answers [17].

The lack of general ET design guidelines, in addition to the scarcity of therapists especially in low- and middle-income countries [112], emphasize the need for computer-aided versions of ET that could be particularly valuable as low-threshold interventions, to treat ADs at a sub-clinical level of symptoms [89]. In this thesis, Reinforcement Learning (RL) [131] has been the algorithm of choice for implementing an AI-assisted version of ET. RL is a branch of Machine Learning (ML) where an intelligent agent takes actions in a dynamic environment in order to maximize a cumulative reward. RL agents do not receive direct instructions regarding which actions to take, but they have to learn the optimal actions through trial-and-error interactions with the environment. RL algorithms have recently made great strides in medical domains, like in the optimization of anti-retroviral therapy in HIV [100], the detection of lung cancer [79], in tailoring anti-epilepsy drugs for seizure control [58], and in determining the best approach to managing sepsis [76]. In the RL framework of this thesis, an RL agent operates as therapist, trying to find an optimal ET protocol for the patient it is currently interacting with. The patients are implemented using computational associative learning models and the RL agent's goal is to maximize fear extinction [89].

In the past, AI methods have been used to optimize associative learning studies [8, 90], to estimate the mental state of other agents [109], and to support the treatment of specific phobias [10, 87], but these advancements have not yet been applied to design efficient, personalized and optimized ET protocols. The limits and possibilities of applying AI, and more specifically RL, in the optimization of ET protocols for fear extinction are therefore largely unknown, which is a gap this thesis is trying to close. To get a clearer understanding of the limitations and possibilities of RL in optimizing EL, this thesis focuses on factors determining the success or failure of applying RL in this context, like the usage of different RL environmental settings, different reward functions and tuning RL model hyperparameters to find ET protocols that lead to robust and efficient fear extinction with the long-term goal of optimized, personalized and widely accessible ET.

The remaining part of this thesis is organized as follows: Chapter 2 describes the research questions addressed in this thesis in more detail. Chapter 3 covers background information, explaining relevant theory and terminology. Topics are ADs, ET, the concept of fear conditioning, different computational associative learning models used as stand-in for the patients, an overview about RL and the RL algorithm used in this thesis, and lastly some information about the RL model hyperparameter tuning process. Chapter 4 covers recent literature related to the usage of AI in the (mental) health sector, discussing relevant findings and future prospects. Chapter 5 outlines implementation details of the models and methods used in the simulations and analysis with a focus on specifications of reward functions, settings for the RL model hyperparameter tuning, and evaluation procedures to determine a successful learning. Chapter 6 outlines specifications of patient model parameters used in the experiments and further experimental details on training, evaluation and hyperparameter tuning. In Chapter 7 the most relevant findings

determining the success or failure of the RL model learning process are summarized. A focus is on suitable patient model parameters, RL model hyperparameters, reward functions, and the transferability of simulation settings from one patient model to another. Chapter 8 further discusses and critically reviews these findings and includes a perspective on potential future extensions. The code related to this project can be found in the following github repository: https://github.com/proj-aether/msc-thesis-tomasiak

2 Research Questions

The aim of this thesis is to investigate the limits and potential of applying RL in the context of optimizing ET protocols. Simulation studies are conducted where computational associative learning models are used as a stand-in for patients and an RL algorithm, representing the therapist, decides on an ET protocol with the goal of robustly extinguishing previously formed negative associations. While the overall aim is to determine the limits and potential of RL in the context of fear extinction, subquestions can be formulated addressing the factors that determine the success or failure of using RL approaches in the considered setting. In particular, the following Subquestions (SQs) have been considered in the course of this thesis:

- SQ1: One parameter of interest is the length of the RL episodes, which is equivalent to the number of extinction trials available for the RL agent to extinguish a patient's previously conditioned fear. While a too small number might not suffice to robustly extinguish negative associations, too many extinction trials can lead to inefficient ET protocols. SQ1 is therefore: How many extinction trials lead to robust and efficient ET protocols in the simulations?

- SQ2: As mentioned above, in the conducted simulations computational associative learning models are used as a stand-in for patients. First, all simulations are conducted with the Rescorla-Wagner (RW) patient model. SQ2 is regarding one of the model's parameters: Which prior for the RW model learning rate captures empirically reasonable speed of human fear learning and supports finding optimized ET protocols in the simulations?

- SQ3: Another factor determining the success or failure of the RL process is the choice on the reward function for the RL algorithm. In this thesis, six different functions are tested, which leads to SQ3: Which factors to consider in the reward function to help finding an optimized ET protocol?

- SQ4: The next SQ is regarding the patient model observation function, which is a mapping of patient's weights associated with presented stimuli to emitted responses expressing their level of fear. Two different observation functions, a linear and a sigmoid one, are used in the simulations and lead to SQ4: How does the patient model observation function influence the RL agent's ability to find an optimized ET protocol?

- SQ5: Another point of investigation is the usage of different associative learning models as a stand-in for patients and the transferability of optimized parameter

settings from one model to another. While all experimental settings are first optimized for the RW model, the chosen parameter values are then passed on to the Kalman Rescorla-Wagner (KRW) and the Dirichlet Process-Kalman Rescorla-Wagner (DP-KRW) model. SQ5 is: Can parameter settings that lead to an optimized ET protocol for one patient model achieve similarly good results when transferred to other related patient models?

3 Background

This chapter presents background knowledge required for understanding the remaining parts of this thesis. First, a brief overview is given about Anxiety Disorders and Anxiety-Related Disorders (ADs) (see Section 3.1) and its most common treatment: Exposure Therapy (ET) (see Section 3.2). Section 3.3 discusses Fear Conditioning (FC), a lab model of how fears are acquired, and Extinction Learning (EL), a lab model of ET. As explained in Chapter 1 and 2, computational associative learning models are used as a stand-in for patients in this thesis' simulations. Section 3.4 outlines the basic concepts of the patient models used. Section 3.5 is about general aspects regarding Reinforcement Learning (RL), with a focus on policy gradient methods and Proximal Policy Optimization (PPO) (see Section 3.5.1). Lastly, Section 3.6 focuses on tuning the RL model hyperparameters.

3.1 Anxiety Disorders and Anxiety-Related Disorders

Anxiety Disorders and Anxiety-Related Disorders (ADs) are the most prevalent psychiatric disorders and associated with a high burden of illness [64, 74]. Up to a third of the population is affected by an AD during their lifetime, often underrecognized, undertreated and following a chronic course [74]. The most common ADs are specific (isolated) phobias with a 12-month prevalence of 10.3% [66], followed by panic disorder with or without agrophobia (PDA) with a prevalence of 6.0%. Social anxiety disorder (SAD), or also called social phobia, has a prevalence of 2.7% and generalized anxiety disorder (GAD) a prevalence of 2.2% [73]. Anxiety disorders mostly occur during midlife and are more common in women and in patients with a family history of anxiety [11]. High rates of comorbidity are observed among various anxiety disorders as well as between anxiety disorders and other mental health conditions [11, 34, 74]. The essential features of anxiety disorders are excessive and enduring fear, anxiety or avoidance of perceived threats and panic attacks [34].

Fear and anxiety are shaped by both (epi)genetic factors and environmental influences [53, 117], which leads to a considerably high rate of inter-individual variation in normal to pathological forms of fear and anxiety [82]. Inter-individual difference are of biological and experiental origin, like e.g., age, sex, hormonal concentrations, brain morphology, genetic polymorphisms, experienced life events and temperamental variables [82]. They may explain the differences in individual fear experience and the variable degree of benefit from anxiety treatments, which underline the need for a personalized approach in the treatment of anxiety disorders [86]. Two main treatment approaches for ADs are medication and psychotherapy, while the combination of both is significantly more beneficial than medication alone [13].

3.2 Exposure Therapy

Exposure Therapy (ET) is an effective approach for treating ADs and has been a mainstay of cognitive behavioral therapy since its inception [62, 99]. During ET, patients are encouraged to systematically approach feared objects, situations, thoughts, sensations, and memories in a safe environment with the aim of reducing their fear [35, 43]. ET procedures can be divided into four primary types [5]:

- In vivo exposure: Direct exposure to a feared object, situation or activity in real life (e.g., handling a snake for patients with fear of snakes).
- Imaginal exposure: Mental confrontation of the anxiety by vividly imagining the feared object or situation (e.g., imagining standing in a crowded mall for patients with fear of crowded places).
- Virtual reality exposure: Using Virtual Reality (VR) technology for exposure, especially useful if in vivo exposure is not practical (e.g., offering a virtual flight including sights, sounds and smells of an airplane for patients with fear of flight).
- Interoceptive exposure: Strategically inducing the somatic symptoms associated with a threat (e.g., running in place in order to speed up heart so that patients with panic disorder learn that this sensation is not dangerous).

The listed ET procedures can be paced in different ways. Specific ET techniques are [5, 72]:

- Graded exposure: An exposure fear hierarchy is created, in which feared objects, activities or situations are ranked according to the level of discomfort. Patients are then gradually exposed, starting with the least difficult to the most difficult exposure.
- Flooding: Using the exposure fear hierarchy to begin exposure with the most difficult tasks. The idea behind flooding is that anxiety levels will be reduced by immersing patients in the situation they fear most, and having them stay in that situation over a period of time [127].
- Systematic desensitization: Combining exposure with relaxation exercises so that patients associate the feared objects, activities or situations with relaxation.

Different mechanisms are thought to underlie the successful applications of ET [5, 72, 96], including:

- Habituation: This theory proposes that after repeated exposure to feared objects or situations, patients experience a natural reduction in fear response due to habituation [57].
- Extinction: A theory that previously learned associations between feared objects or situations and negative outcomes are weakened through exposure [115, 136].

- Self-efficacy: This theory suggests that exposure can help show patients to be capable of confronting feared objects or situations and managing feelings of anxiety. The self-efficacy theory focuses more on increasing mastery over a situation and capabilities to cope with fear than on reducing a fear response directly [14].
- Emotional processing: A theory stating that during exposure patients can learn to attach new, more realistic beliefs about feared objects or situations [37, 44]. In 2014 the inhibitory learning theory has been introduced [35], purporting that the initial conditioned association between a feared object or situation and a negative outcome is not erased, but instead a new association is learned, competing with the former response.

Experimental studies regarding the acquisition of fear suggest that the development of ADs can be modeled by FC [55]. ET on the other hand can be modeled by Extinction Learning (EL), where aversive associations are extinguished after they have been experimentally induced [36]. Details regarding FC and EL are outlined in the following section.

3.3 Fear Conditioning and Extinction Learning

Fear Conditioning (FC) is a form of classical Pavlovian conditioning [103], where a neutral stimulus, Conditioned Stimulus (CS), is repeatedly paired with a biologically significant reinforcer, Unconditioned Stimulus (US), to study how animals' predictions of reinforcement develop with experience of various CS/US patterns. The US is a stimulus that triggers a naturally occurring response, the Unconditioned Response (UR). When the US is repeatedly presented with the CS, the CS evokes a similar response as the US, called Conditioned Response (CR). The CR is a behavioral, physiological measure used to assess predictions in the experiments as it is thought to directly reflect the expectation of a US. An example is Pavlov's experiment [103], where the repeated pairing of a bell (CS) with food (US) gradually leads to animals salivating (CR) in response to the bell alone. In this example the UR and the CR are the same behaviour (salivation) but they are named differently as they are produced by different stimuli (the US and the CS, respectively).

While repeatedly presenting the CS in combination with the US leads to a conditioned fear, the process of extinguishing this fear is called Extinction Learning (EL). Extinction is a complex phenomenon that has resisted simple explanation since its first characterization by Pavlov [103]. In its most basic variant, extinction is achieved by repeatedly and reliably presenting the CS without US, which results in a decline in CRs. Extinction is not the same as forgetting as it requires exposure to the CS in absence of the US as opposed to the simple passage of time. Generally speaking, any procedure that compromises the predictive value of the CS regarding the occurrence of the US will result in some reduction of the CR to the CS. The most obvious procedure leading to a reduced CR is the one mentioned above, where a CS, that was previously paired with a US, is repeatedly presented without further US delivery. However, extinction occurs under a variety of training protocols, like reinforcing the CS on a smaller percentage of its presentations than during acquisition or continuing to pair the CS with the US on 100% of its presentations

but lowering the intensity of the US compared to acquisition. A critical determining factor for fear extinction to occur seems to be a violation of the predicted and therefore expected connection between the CS and the US, learned from the trials in earlier stages of the training. [96]

In order to assess aversive associations, CRs measured immediately after FC (fear acquisition tests), immediately after EL (fear extinction tests), and follow-up assessments (extinction retention tests or Return of Fear (RoF) tests) are especially interesting. While the tests after fear acquisition assess the success of FC, the tests after the EL phase measure the success of the applied EL protocol. The RoF tests assess the lasting of fear extinction under various aspects like the passage of time or context changes, with returning of fear being termed reinstatement, renewal, and spontaneous recovery. Reinstatement refers to a reappearance of extinguished fear responses after presenting the US several times alone subsequent to extinction training. Renewal happens in the case of reappearance of extinguished CRs when tested in a different context than during extinction training. Spontaneous recovery describes the reappearance of extinguished CRs with the passage of time after extinction training in absence of any further explicit training. [81, 96]

Fear conditioning experiments often use two conditioned stimuli: CS+ (a CS for which a negative association is acquired through pairing it with an aversive US during FC) and for comparison reasons CS- (a CS for which no negative association is acquired during FC).

3.3.1 Standard Fear Conditioning Paradigm

Summarizing the above, the standard FC paradigm can be divided into three stages [22], which are outlined in the following and schematically represented in Figure 3.1.

- **Stage 1: Fear Acquisition**
  During fear acquisition, fear responses are acquired for a CS (referred to as CS+) through repeatedly presenting the CS with an aversive US. These negatively reinforced CS presentations are interleaved with presentations of another CS (referred to as CS-), which is repeatedly presented without the US. The CS- serves as reference by showing a patient's response behavior towards non-reinforced CSs. This stage can be seen as a lab model of FC.

- **Stage 2: Fear Extinction**
  During stage 2, fear extinction protocols are carried out, leading to diminishing fear responses. The most basic extinction protocol is presenting the previously negatively reinforced CS+ (and for comparison the non-reinforced CS-) repeatedly without the US. This stage can be seen as a lab model of ET.

- **Stage 3: Extinction Retention**
  When re-exposed to the CS+ after extinction training, fear responses may return or be inhibited. A Return of Fear (RoF) test functions as a measure of retained fear, by assessing the emitted CRs to several presentations of the CS+ (and for comparison CS-) without the US.

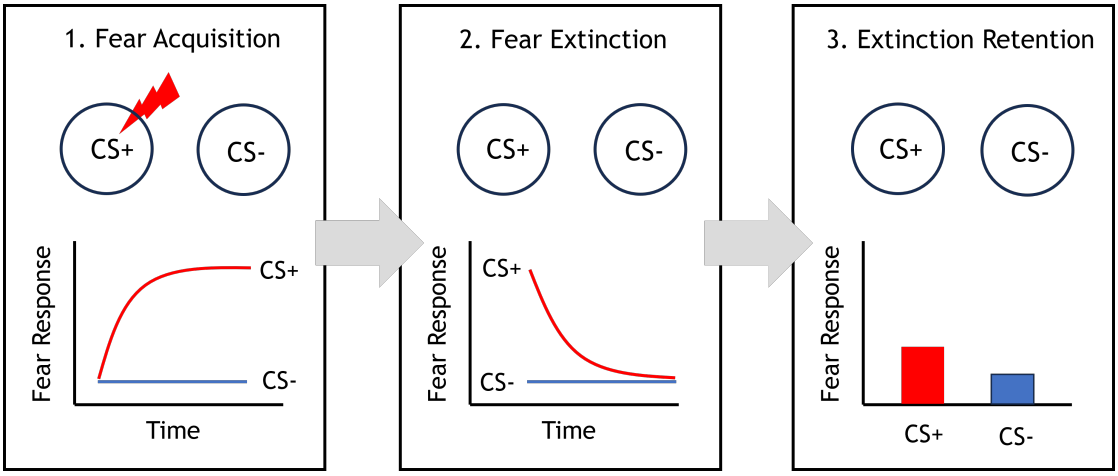


Figure 3.1: Schematic illustration of the three fear conditioning stages reconstructed from [22]. In stage 1 (fear acquisition) a CS (CS+) is paired with the US, leading to increasing fear responses. Another CS (CS-) is repeatedly presented without the US, serving as a reference in all stages. During stage 2 (fear extinction) the association between CS+ and US is extinguished by presenting the CS+ repeatedly without negative reinforcement. Stage 3 (extinction retention) assesses retained fear responses to the CS+. The stages are performed sequentially, while CS+ and CS- presentations are interleaved within each stage.

3.4 Computational Associative Learning Models

In this thesis' experiments, cognitive models of associative learning are used as a stand-in for patients. Cognitive modeling, within the realm of cognitive science, involves creating computational models to simulate mental processes and human problem-solving [25].

The three computational associative learning models used in this thesis aim to describe the mechanisms of conditioning phenomena. They are the Rescorla-Wagner (RW) model, the Kalman Rescorla-Wagner (KRW) model and the Dirichlet Process-Kalman Rescorla-Wagner (DP-KRW) model, outlined in the following sections. The Dirichlet Process-Kalman Rescorla-Wagner (DP-KRW) model is a latent cause model [31, 33] and is discussed in terms of the general idea behind latent cause models in this section, while later in Section 5.2.3 model-specific details are described.

3.4.1 Rescorla-Wagner Model

The Rescorla-Wagner (RW) model [115] is a model in classical conditioning which conceptualizes learning in terms of associations between CS and US. It was very influential in the history of animal learning models by building the foundation for modern associative learning theory and serving as cornerstone for more sophisticated models [48, 105, 134]. Despite some shortcomings [92], the RW model represents a major advantage over previous

models by explaining some experimental phenomena in classical conditioning like e.g. blocking (a phenomenon where a previous CS conditioning process prevents conditioning to another CS).

The main idea of the RW model is that animals only learn in surprising situations and therefore when events violate their expectations. Assuming that a trial n is described by a vector $x_n = (x_{n,1}, x_{n,2}, ..., x_{n,d})^T$ where $x_{n,i} = 1$ if CS_i (the i^{th} component of a compound CS) is presented and $x_{n,i} = 0$ otherwise. For each trial n, the individual CSs' associative strengths are stored in a weight vector w_n, which is used to calculate an expected US. Subtracting the expected US from the actual US yields a prediction error δ_n, which is used to update the associative strengths applying the following update rule:

$$w_{n+1} = w_n + \alpha\delta_n x_n \tag{3.1}$$

with learning rate parameter α influencing the magnitude of the weight update. The value for α can be chosen depending on the experiment and is by definition $\in [0, 1]$ [115]. Further implementation details of the RW model are outlined in Section 5.2.1.

3.4.2 Kalman Rescorla-Wagner Model

Interpreting the RW model in a probabilistic way, it is a maximum likelihood estimator representing the single most likely weight vector [48]. Bayesian theories of learning suggest that learners estimate not only the strength of associations but also report uncertainty about these associations [6, 49], which necessitates to consider models that explicitly represent this uncertainty. The Kalman Rescorla-Wagner (KRW) model is a Bayesian model of learning which represents uncertainty by formulating a posterior distribution over hypotheses based on available data. For associative learning, Bayes' rule specifies the posterior distribution as follows:

$$p(w_n|x_{1:n}) \propto p(x_{1:n}|w_n)p(w_n) \tag{3.2}$$

where index $1 : n$ denotes all trials from 1 to n, and definitions for trial, w_n and x_n are the same as for the RW model described in the above Section 3.4.1. $p(w_n|x_{1:n})$ represents the posterior distribution over the associative strength w_n after the presentation of sequence $x_{1:n}$, $p(x_{1:n}|w_n)$ represents the likelihood function, which quantifies the probability of being presented to sequence $x_{1:n}$ given the associative strength w_n, and $p(w_n)$ represents the prior distribution over the associative strength w_n, reflecting the initial belief about the strength of the association.

Under a specified linear-Gaussian dynamical system (LDS) (see Section 5.2.2), the posterior is Gaussian with mean w_n and covariance matrix $\sum_n$, which both get updated after each trial. For updating the associative strength vector w_n, the KRW model makes use of the so-called *Kalman gain*, introducing the learner's uncertainty. The update equation looks like the following:

$$w_{n+1} = w_n + K_n\delta_n \tag{3.3}$$

with Kalman gain K_n, which is dynamically adapted according to Bayes' rule and replaces the learning rate α in the RW model. For mathematical details used in the implementation see Section 5.2.2.

The KRW model explains some of the phenomena that are not captured by the RW model [48]. Although there is considerable evidence for the existence of error-driven learning and stimulus competition, several violations are documented [92]. One example is the latent inhibition [85], where a stimulus alone presented prior to pairing it with reward, retards the acquisition of a stimulus-reward association. With initial associative strengths set to 0 in the RW model, the prediction error is 0 during pre-exposure leading to no associative learning in this phase. The RW model is therefore not able to capture the intuition of CS pre-exposure retarding the acquisition of a stimulus-reward association, while the KRW can model this latent inhibition by attenuating posterior uncertainty and therefore decreasing the Kalman gain after repeated CS presentation.

Furthermore, the KRW model provides an explanation for a range of phenomena, where compound stimulus training causes the acquisition of negative covariance between the stimulus elements. One example is backward blocking [122], where a compound of two stimuli is paired with reward and then one of the two stimuli alone is paired with reward, leading to a reduced responding of the other stimulus alone. The model learns during compound training phase that the CSs' weights must sum to 1 and therefore one weight being large necessitate the other one being small, mathematically encoded as negative off-diagonals in the covariance matrix Σ_n [48].

3.4.3 Latent Cause Models

One of the main weaknesses of the RW model is its prediction that presenting the CS repeatedly without US will erase the CS-US association formed during acquisition [50]. The assumption that negative associations are unlearned, although shared by other models [104], is meanwhile widely accepted to be wrong [38, 47]. Bouton et al. [21] reviewed a range of conditioning phenomena in a post-extinction test phase where fear responses reappeared following apparent extinction. This phenomenon is known as spontaneous recovery, where increasing the time between extinction and test is sufficient to increase responding to the CS, for which negative associations have been extinguished [103, 114].

The recovery phenomena can be explained by latent cause models [31, 32, 33], according to which a subject's internal model consists of basic building blocks referred to as *latent causes*, which are hidden from observation. Spontaneous recovery can occur because acquisition trials are associated with a different latent cause than the extinction trials, preventing the extinction of the original aversive associations [33, 49].

In 2010, Gershman et al. [49] introduced the latent cause theory of classical conditioning. They explain that subjects combine their a priori beliefs about how the world is structured with current observations to make conclusions about how CSs and USs are linked, and to make predictions about future occurrences of a US. The term observation refers to the set of features (CS, US, optionally context,...) presented on a particular trial, and a trial describes a single instance of presenting this set of features.

One assumption is that animals use Bayes' rule to combine prior beliefs and the likelihood of an observation in order to form a posterior belief

$$P(\text{cause}|\text{data}) \propto P(\text{data}|\text{cause})P(\text{cause}) \tag{3.4}$$

where "data" refers to the observation in a trial and "cause" refers to the hypothesized latent cause underlying the current observation. For each trial there is a posterior distribution over its membership in the different latent causes, and therefore in the Dirichlet Process-Kalman Rescorla-Wagner (DP-KRW) model used in this thesis' simulations, each latent cause is associated with a different set of weights w representing the associations between CSs and USs, while the total US prediction is done by averaging across causes. As the latent cause itself is not observed, the subject must rely on its inference about which trials were caused by which latent causes. Gershman and colleagues impute to the subjects a set of probabilistic assumptions about their surrounding environment, which have an influence on the computational implementation of the latent cause model [51]:

1. Each trial is caused by one latent cause.
2. Each latent cause has some characteristic probability of emitting observed features (CS, US, etc.).
3. If all else is equal, a prolific latent cause (i.e., one that has caused many trials) is more likely to cause another trial.
4. There is some small probability that the current trial results from a completely new latent cause not having generated any observations yet.

According to this assumptions, the surrounding environment tends to change slowly over time, while jumps between different modes (latent causes) only occasionally occur (for mathematical details see Section 5.2.3). Furthermore, there is a reasoning backward from observations to latent causes using Bayes' rule (see Figure 3.2).

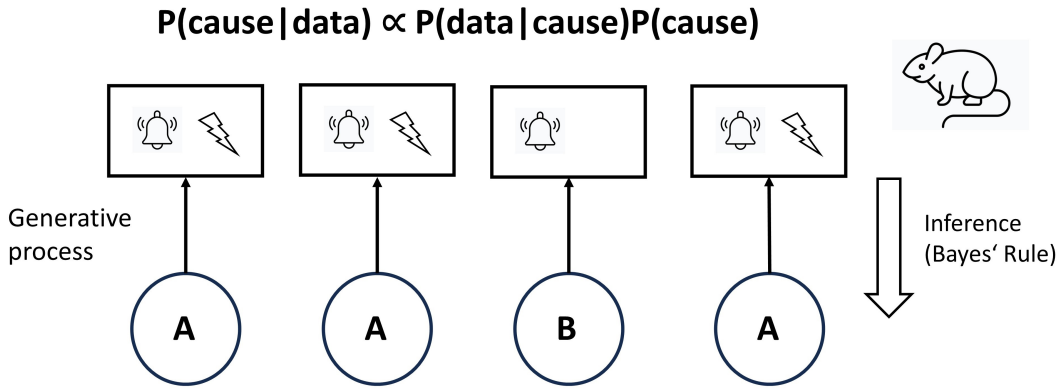


Figure 3.2: Schematic representation of the latent cause theory reproduced from [51]. The boxes represent observations on single trials and the circles represent latent causes. Observations are assumed to be generated by latent causes and hence the upward arrows denote this generative process. As the animal cannot observe the latent cause, it must infer it using Bayes' rule, indicated by the downward arrow.

Once the latent cause is inferred, the animal can generate a CR based on whether the characteristic emission probabilities of the latent cause suggest the appearance of a US. When sensory data change gradually over time, an existing memory (the current representation of the environment) is updated, while abrupt changes lead to the formation of new memories [51] (for mathematical details see Section 5.2.3).

3.5 Reinforcement Learning

The aim of this thesis is to investigate the limits and potential of applying Reinforcement Learning (RL) [131] in the context of optimizing ET protocols. An RL task is formally described by means of a Markov Decision Process (MDP) [108, 131], which is a tuple $\langle S, A, R, P, \gamma \rangle$ where

- $\mathcal{S}$ is a (finite) set of states;
- $\mathcal{A}$ is a (finite) set of actions;
- R is a reward function: $R : S \times A \to \mathbb{R}$
- P is a transition function: $P : S \times A \times S \to [0, 1]$
- γ is a discount factor: $\gamma \in [0, 1]$.

The Markov property is assumed, which states that the state of the next timestep S_{t+1} depends only on the state of the current timestep S_t and not on previous ones:

$\mathbb{P}[S_{t+1}|S_1, A_1...S_t, A_t] = \mathbb{P}[S_{t+1}|S_t, A_t]$. The probability of each pair of next state and reward (s', r) for any state-action pair (s, a) is given by [30]:

$$p(s', r|s, a) = \mathbb{P}[S_{t+1} = s', R_{t+1} = r|S_t = s, A_t = a] \tag{3.5}$$

where small letters, e.g., s, a, and r, represent a state, action or reward respectively and capital letters, e.g., S_t, A_t, and R_t, represent a state, action or reward at timestep t [131]. In the standard reinforcement learning setting an agent interacts with an environment over a number of discrete timesteps. At each timestep t the agent receives a state $S_t \in \mathcal{S}$ on which' basis it selects an action $A_t \in \mathcal{A}$ according to a policy π. As a consequence of its action, one timestep later the agent receives a numerical reward R_{t+1}, and finds itself in a new state S_{t+1}. This process continues until a terminal state is reached after which the process restarts (see Figure 3.3). Denoting the sequence of rewards after timestep t as $R_{t+1}, R_{t+2}, ...$, the objective is to maximize the accumulated, discounted return from timestep t defined as:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{3.6}$$

where $\gamma \in [0, 1]$ is the discount rate. The discount rate determines how strongly future rewards are taken into account, where a reward received k time steps in the future is worth only γ^{k-1} times what it would be worth when received immediately.

A policy is a set of stimulus-response rules, which map each state of the environment to an action. A policy π is therefore a probability distribution over actions given states, i.e.

$$\pi(a|s) = \mathbb{P}[A_t = a|S_t = s]. \tag{3.7}$$

Given the policy π and the return G_t, it is possible to specify two value functions related to the expected return, the state-value function and the action-value function, which are described in the following.

The state-value function $v_\pi(s)$ is the expected return starting from state s and following policy π, i.e.

$$v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s \right] \tag{3.8}$$

where $\mathbb{E}_\pi[\cdot]$ denotes the expected value given that the agent follows policy π, and t is any timestep.

The action-value function $q_\pi(s, a)$ is the expected return from state s, selecting action a, and afterwards following policy π:

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s, A_t = a \right]. \tag{3.9}$$

It is possible to define the optimal value functions which specify the best performance in terms of returns. The optimal state-value function $v_*(s)$ is the maximum state-value function over all policies:

$$v_*(s) = \max_\pi v_\pi(s). \tag{3.10}$$

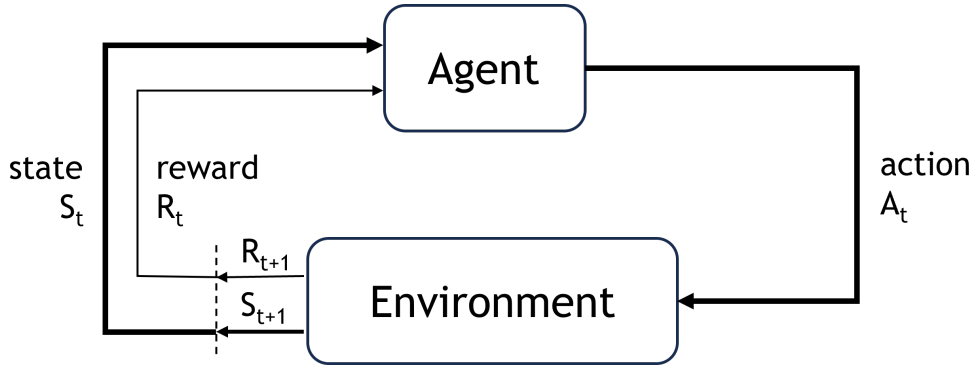


Figure 3.3: The classical RL framework reproduced from [131]. The agent selects an action A_t at state S_t and receives a corresponding reward R_{t+1}. The agent's objective is to choose actions that maximize its cumulative reward over a sequence of transitions.

The optimal action-value function $q_*(s, a)$ is the maximum value function over all policies:

$$q_*(s, a) = \max_{\pi} q_\pi(s, a). \tag{3.11}$$

RL algorithms can be divided into two main categories: model-free and model-based RL [131]. While model-free RL algorithms directly optimize the policy based on the gathered interaction experience, model-based RL algorithms additionally learn a model of the environment that predicts state transitions and rewards [131].

This thesis' simulations have been conducted using a model-free, actor-critic RL approach. Besides actor-critic based approaches, there are policy-based and value-based methods in model-free RL [131]. While value-based methods select actions based on their estimated action values $q_\pi(s, a)$, policy-based methods aim to find a parameterized policy π_θ which is used for action selection. Actor-critic methods combine the strengths of policy-based and value-based methods. They use two neural networks: the actor, deciding which action to take, and the critic, evaluating the action taken by the actor [56].

The model-free RL algorithm used for training the models in this thesis is Stable Baselines3's [111] RecurrentPPO with MlpLstm policy. The MlpLstm policy object implements actor and critic, using Long Short-Term Memories (LSTMs) [9] with a Multilayer Perceptron (MLP) [61] feature extraction.

3.5.1 Policy Gradient Methods

Stable Baselines3's [111] RecurrentPPO implements recurrent policies for the Proximal Policy Optimization (PPO). PPO is a policy gradient method, which will first be explained on a general level in this section, followed by a more detailed description of the PPO implementation. A trajectory of the form $\tau = (S_1, A_1, R_2, S_2, A_2, ..., S_T, A_T, R_{T+1})$ gives the cumulative reward defined as $G(\tau) = \sum_{t=0}^{T} \gamma^t R_{t+1}$. The goal of RL is to find a policy π with parameters θ (π_θ) that maximizes $G(\tau)$ in expectation. This goal can be

formulated in terms of a performance measure $J(\theta)$, i.e.

$$J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[G(\tau)], \tag{3.12}$$

where $\tau \sim \pi_\theta$ indicates the trajectory τ generated following policy π_θ. The policy parameters θ dictate the sampling probability of actions a, leading to a stochastic policy $\pi_\theta(a|s)$. θ can then be updated in a direction that yields higher rewards, which is the mechanism underlying all policy gradient methods. The problem of finding the optimal policy parameters θ^* can be summarized as:

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\ \mathbb{E}_{\tau\sim\pi_\theta}[G(\tau)] = \underset{\theta}{\operatorname{argmax}}\ J(\theta). \tag{3.13}$$

Policy gradient methods aim to find θ^* with a stochastic gradient ascent algorithm based on an estimate of $\nabla_\theta J(\theta)$ [18]. The most commonly used gradient estimator has the following form [119]:

$$\nabla_\theta J(\theta) \approx \hat{\mathbb{E}}_t[\nabla_\theta \log \pi_\theta(A_t|S_t)\hat{A}_t] \tag{3.14}$$

where the expectation $\hat{\mathbb{E}}_t[...]$ denotes the empirical expectation over a finite set of samples and $\hat{A}_t$ is an estimator of the advantage function at timestep t. $\hat{A}_t$ can be mathematically defined in various ways depending on the RL algorithm and the problem setting. One style of policy gradient implementation was popularized by Mnih et al. [93] and uses the following estimator of the advantage function

$$\hat{A}_t = -v_\pi(S_t) + R_t + \gamma R_{t+1} + \ldots + \gamma^{T-t+1} R_{T-1} + \gamma^{T-t} v_\pi(S_T) \tag{3.15}$$

where t specifies the time index in $[0, T]$, within a given trajectory segment with length T, $v_\pi(S_t)$ is the state-value of state S_t (see Equation (3.8)), R_t is the reward at timestep t, and γ serves as discount factor controlling the importance of future rewards. The policy gradient method used in this thesis' simulations is the Proximal Policy Optimization (PPO), which is described in more detail in the following section.

Proximal Policy Optimization (PPO)

The Proximal Policy Optimization (PPO) is a policy gradient method used in RL to train an agent's policy network (see Algorithm 1). It has been developed by Schulman et al. in 2017 [119] who found that it shows superior overall performance and simpler implementation compared to other prevalent policy gradient methods. As described above, policy gradient methods try to find optimal policy parameters maximizing a specified performance measure (see Equation (3.13)). PPO updates policies according to

$$\theta_{k+1} = \underset{\theta}{\operatorname{argmax}}\ \mathbb{E}_{\tau\sim\pi_\theta}[L(s, a, \theta_{\text{old}}, \theta)] \tag{3.16}$$

where k indicates the current PPO iteration and $L(s, a, \theta_{\text{old}}, \theta)$ (from now on abbreviated as $L(\theta)$) is a loss function with a state action pair (s, a), θ_{old} are the policy parameters

before the update and θ the new policy parameters. The main objective Schulman and colleagues propose is

$$L_t^{\text{CLIP}}(\theta) = \min\left(\rho_t(\theta)\hat{A}_t, \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right) \tag{3.17}$$

where $\rho_t(\theta)$ is the probability ratio computed as the probability of taking action A_t at state S_t in the current policy π_θ divided by the probability of action A_t at state S_t in the previous policy $\pi_{\theta_{\text{old}}}$

$$\rho_t(\theta) = \frac{\pi_\theta(A_t|S_t)}{\pi_{\theta_{\text{old}}}(A_t|S_t)}. \tag{3.18}$$

clip() is a clipping function with hyperparameter ϵ, regulating how much the new policy can deviate from the old policy while still profiting the objective. This concept is explained on a more intuitive level after some further definitions. Schulman et al. [119] use a truncated version of Mnih et al.'s [93] generalized advantage estimation (see Equation (3.15))

$$\hat{A}_t = \delta_t + \gamma\lambda\delta_{t+1} + (\gamma\lambda)^2\delta_{t+2} + \ldots + (\gamma\lambda)^N\delta_{t+N}, \tag{3.19}$$

$$\text{where} \quad \delta_t = (R_t + \gamma v_\pi(S_{t+1}) - v_\pi(S_t)), \tag{3.20}$$

which reduces to Equation (3.15) if the smoothing factor $\lambda = 1$. Equation (3.17) can be expressed in the following simplified and arguably more intuitive form (for derivation see [2]):

$$L_t^{\text{CLIP}}(\theta) = \min\left(\rho_t(\theta)\hat{A}_t, g(\epsilon, \hat{A}_t)\right) \tag{3.21}$$

where

$$g(\epsilon, \hat{A}_t) = \begin{cases} (1+\epsilon)\hat{A}_t & \text{if } \hat{A}_t \geq 0 \\ (1-\epsilon)\hat{A}_t & \text{otherwise} \end{cases} \tag{3.22}$$

The two cases of the advantage being positive and the advantage being negative are outlined in the following and visually presented in Figure 3.4.

Advantage is positive: If the advantage $\hat{A}_t$ for a state-action pair (S_t, A_t) is positive, the contribution to the objective $L_t^{\text{CLIP}}(\theta)$ reduces to

$$L_t^{\text{CLIP}}(\theta) = \min\left(\rho_t(\theta), (1+\epsilon)\right)\hat{A}_t. \tag{3.23}$$

Due to the positive advantage, the objective will increase if the action becomes more likely, i.e. if $\pi_\theta(A_t|S_t)$ in $\rho_t(\theta)$ increases. The min() in Equation (3.23) puts a limit on how much the objective can increase. Once $\pi_\theta(A_t|S_t) > (1+\epsilon)\pi_{\theta_{\text{old}}}(A_t|S_t)$ this term gets limited to $(1+\epsilon)\hat{A}_t$, which has the effect that it is not profitable if a new policy $\pi_\theta(A_t|S_t)$ moves far from the old policy $\pi_{\theta_{\text{old}}}(A_t|S_t)$.

Advantage is negative: If the advantage for a state-action pair (S_t, A_t) is negative, the contribution to the objective $L_t^{\text{CLIP}}(\theta)$ reduces to

$$L_t^{\text{CLIP}}(\theta) = \min\left(\rho_t(\theta), (1-\epsilon)\right)\hat{A}_t. \tag{3.24}$$

Due to the negative advantage, the objective will increase if the action becomes less likely, i.e. if $\pi_\theta(A_t|S_t)$ in $\rho_t(\theta)$ decreases. The min() in Equation (3.24) puts a limit on how

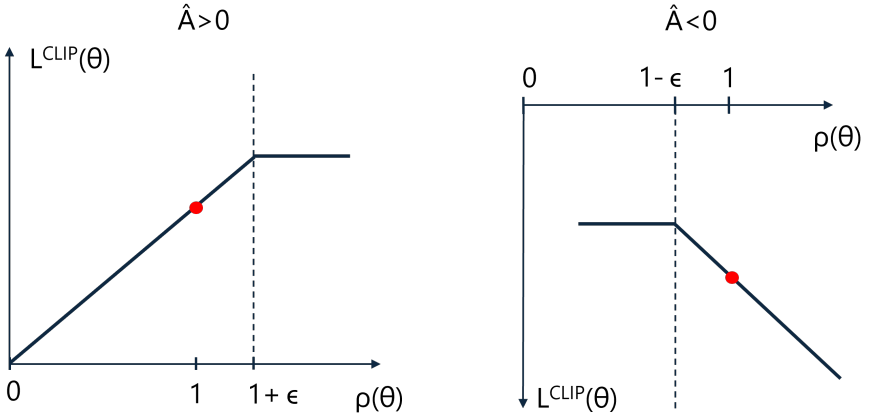


Figure 3.4: Plots showing one term (i.e., a single timestep) of the surrogate function $L^{\text{CLIP}}(\theta)$ as a function of the probability ratio $\rho(\theta)$, for positive $\hat{A}$ (left) and negative $\hat{A}$ (right). The red circle on each plot shows the starting point for the optimization, i.e., $\rho(\theta) = 1$. Note that the probability ratio $\rho(\theta)$ is clipped at $1 - \epsilon$ or $1 + \epsilon$ depending on whether the advantage is positive or negative. Figure reproduced from [119].

much the objective can increase. Once $\pi_\theta(A_t|S_t) > (1-\epsilon)\pi_{\theta_{\text{old}}}(A_t|S_t)$ this term gets limited to $(1-\epsilon)\hat{A}_t$, which has the effect that it is not profitable if a new policy $\pi_\theta(A_t|S_t)$ moves far from the old policy $\pi_{\theta_{\text{old}}}(A_t|S_t)$.

Figure 3.4 shows the two cases for one single timestep in $L^{\text{CLIP}}(\theta)$ showing that the probability ratio $\rho(\theta)$ is clipped at $1+\epsilon$ or $1-\epsilon$ depending on whether the advantage $\hat{A}$ is positive or negative.

If using a neural network architecture that shares parameters between the policy and value function, Schulman and colleagues [119] propose a loss function consisting of a policy surrogate $L_t^{\text{CLIP}}(\theta)$, a value function error term $L_t^{\text{VF}}(\theta)$ and an optional entropy bonus $H(\pi_\theta(\cdot|s_t))$ used to encourage exploration during training

$$L_t^{\text{CLIP+VF+S}}(\theta) = \hat{\mathbb{E}}_t\Big[L_t^{\text{CLIP}}(\theta) - c_1 L_t^{\text{VF}}(\theta) + c_2 H(\pi_\theta(\cdot|s_t))\Big] \tag{3.25}$$

with coefficients c_1 and c_2. The state entropy $H(\pi_\theta(\cdot|s_t))$ is based on the probability distribution of action selections to measure the uncertainty of the action selection for a certain state S_t

$$H(\pi_\theta(\cdot|S_t)) = -\sum_{a \in A} \pi(a|S_t) \log \pi(a|S_t) \tag{3.26}$$

where $\sum_{a \in \mathcal{A}}$ denotes all possible actions a in the set of actions $\mathcal{A}$ [139]. The value function error term $L_t^{\text{VF}}(\theta)$ is a squared-error loss computed as

$$L_t^{\text{VF}}(\theta) = (v_\pi(S_t) - v_t^{targ})^2 \tag{3.27}$$

where $v_\pi(S_t)$ is the value of state S_t following policy π and v_t^{targ} is the target value for state S_t [119, 132]. The PPO algorithm that uses fixed-length trajectory segments is

shown in Algorithm 1. Each iteration an actor collects T timesteps of data on which the surrogate loss is constructed. This surrogate loss is optimized with minibatch Stochastic Gradient Descent (SGD) (or for better performance, Adam [75]), for K epochs.

Algorithm 1 PPO, Actor-Critic Style [119]

for iteration $= 1, 2, \ldots$ **do**
  Run policy $\pi_{\theta_{\text{old}}}$ in environment for T timesteps
  Compute advantage estimates $\widehat{A}_1, \ldots, \widehat{A}_T$
  Optimize surrogate L wrt θ, with K epochs and minibatch size $M \leq T$
  $\theta_{\text{old}} \leftarrow \theta$
end for

3.6 Hyperparameter Tuning

Hyperparameter tuning is a procedure used for optimizing model performance by choosing optimal hyperparameters for a learning algorithm. Bayesian optimization has emerged as an efficient framework in the context of hyperparameter tuning, outperforming other conventional methods such as grid search and random search [121, 125]. In this thesis, the hyperparameter optimization framework Optuna [3] has been used for tuning RL model hyperparameters with its default TPESampler, outlined in more detail the next section.

3.6.1 Tree-Structured Parzen Estimator Sampler

The Tree-Structured Parzen Estimator (TPE) sampler [16] is an optimization algorithm for hyperparameter tuning using Bayesian optimization, which starts by selecting a subset of hyperparameters and sorting them based on their Expected Improvement (EI) [69]. EI is the acquisition function used to find an optimal hyperparameter configuration x^* and is defined as

$$\text{EI}_{y^*}[x|D] := \int_{-\infty}^{\infty} (y^* - y) p(y|x, D) dy. \tag{3.28}$$

with the hyperparameter configuration x, the observation of the objective function y, the set of N observations $D := (x_n, y_n)_{n=1}^N$ and a specified threshold y^* chosen by the TPE algorithm. The TPE sampler is restricted to a tree-structured search space, where some leaf variables (e.g. the number of hidden units in a specific layer) are only well-defined when node variables (e.g. a discrete number of layers to use) take particular values. The TPE models $p(y|x, D)$ using the following assumption:

$$p(x|y, D) = \begin{cases} p(x|D^{(l)}) & (y \leq y^{\upsilon}) \\ p(x|D^{(g)}) & (y > y^{\upsilon}) \end{cases} \tag{3.29}$$

where $D^{(l)}$ and $D^{(g)}$ are a better and a worse group in D, $\upsilon \in (0, 1]$ is a top quantile used to specify the better group $D^{(l)}$, $y^{\upsilon} \in \mathbb{R}$ is the top-υ quantile objective value in D computed each iteration based on the number of observations $N(= |D|)$ and $p(x|D^{(l)})$

and $p(x|D^{(g)})$ are the probability density functions of the better group and the worse group built by Kernel Density Estimators (KDEs). The set of observations D is assumed to be sorted by y_n such that $y_1 \leq y_2 \leq \dots y_N$. This way, the better and the worse group are obtained as $D^{(l)} = (x_n, y_n)_{n=1}^{N^{(l)}}$ and $D^{(g)} = (x_n, y_n)_{N^{(l)}+1}^{N}$ where $N^{(l)} = \upsilon N$.

To maximize improvement, the goal is to find points x_n with high probability under $D^{(l)}$ and low probability under $D^{(g)}$. On each iteration, the algorithm returns the candidate x^* with the greatest EI. This process continues until the budget is depleted, at which point the optimal hyperparameters will be returned [16, 137]

4 Related Work

Driven by the increasing availability of data and the further development of computational models and algorithms, the application of AI techniques in healthcare has significantly increased in the past decade [40, 59, 67, 102]. This chapter discusses recent research regarding the usage of AI techniques in the healthcare sector, with a focus on RL and ADs. Section 4.1 discusses some applications of RL in the general and personalized healthcare sector. In Section 4.2 the focus shifts towards mental health, where first applications of AI techniques in the context of different mental disorders are discussed, and later in Section 4.2.1 the emphasis is on using AI in the treatment of ADs. Lastly, Section 4.3 covers how cognitive models can support AI systems, e.g., in the prediction of human behavior.

4.1 RL in Healthcare

RL approaches have been successfully applied in a number of healthcare domains to date [30, 130, 141], e.g., in the optimization of anti-retroviral therapy in HIV [100], in the detection of lung cancer [79], in tailoring anti-epilepsy drugs for seizure control [58], and in determining the best approach for managing sepsis [76]. The application domains of RL in the healthcare sector can be categorized into three main types: (1) improving precision medicine (2) automated medical diagnosis, and (3) other general domains such as drug discovery and development, health management, health resources allocation, and optimal process control [1, 141]. For this thesis' topic of finding an optimal, personalized ET protocol, research regarding the usage of RL in the context of improving precision medicine is of special interest and therefore reviewed in the following.

In the field of precision medicine, Dynamic Treatment Regimes (DTRs) are especially well suited for RL approaches, as their design can be viewed as a sequential decision making problem [30, 135]. acrshortdtrs are composed of a sequence of decision rules to determine the course of action (e.g., drug dosage, treatment type, or reexamination timing) according to the current health status and prior treatment history of an individual patient. RL approaches for optimizing DTRs have been developed against a variety of illnesses. The authors of [124] have implemented an RL-based approach to learn an optimal DTR that minimizes the patient's schizophrenia symptoms. Komorowski and colleagues [76] propose an RL approach for individualized treatment of sepsis, demonstrating that an RL clinician's selected treatment scores on average higher than selected from human clinicians. Zhao et al. [142] present a value-based RL framework to discover individual treatment strategies for lung cancer and showed that their procedure can extract optimal

strategies from a single set of patient trajectories. Yu and colleagues [140] used a policy gradient method to get better DTRs in Human Immunodeficiency Virus (HIV) management. The authors found that more effective and robust DTRs for HIV can be derived by incorporating causal factors between choices of anti-HIV medicines and the related treatment outcomes.

In 2018 Petersen et al. [106] investigated whether an adaptive, personalized multi-cytokine mediation is an effective approach for lowering patient mortality for sepsis. The authors used RL-based simulations to train a treatment policy in which systemic patient measurements are used in a feedback loop to inform future treatments. They identified a policy achieving 0% mortality on the patient parameterization it was trained on, and reduced mortality over 500 randomly selected patient parameterizations compared to their baseline mortalities. These results suggest that RL-based adaptive, personalized multi-cytokine mediation therapy could be a promising approach for treating sepsis.

Lowery et al. [84] implemented an actor-critic RL algorithm to personalize anesthesia via propofol infusion control. The algorithm learns a generic effective control policy based on patients' clinical data and then tunes the degree of anesthesia in a personalized way. The authors show that the policy can be learned and improved in very few simulated operations, making RL a promising candidate for supporting anesthesia control throughout an operation based on patients' changing responses.

4.2 AI in Mental Health

A recent review on the application of AI in the discipline of mental health shows the potentials of AI in redefining mental illnesses more objectively, identifying them at an earlier stage, and personalizing treatments based on an individual's unique characteristics [54]. The authors reviewed 28 studies that predicted or classified mental health illnesses using AI. They found that depression is the most common mental illness investigated (18/28, e.g., [4, 116]), followed by schizophrenia and other psychiatric illnesses (6/28, e.g., [71, 98]), suicidal ideation or attempts (4/28, e.g., [29, 42]) and general mental health (1/28, [52]), while the most common AI techniques used in the studies were supervised ML techniques (23/28).

But also RL is subject of discussion, when it comes to the potentials of AI in gaining understanding of mental health related disorders, e.g., by using RL to model brain functionalities involved in psychiatric and neurological disorders [88]. Chen et al. [28] reviewed the application of RL to model learning and behavior of patients with depression, showing that RL approaches have contributed substantially to gain more insights into the underlying mechanism and specific processes of depression.

The following section focuses on literature regarding the usage of AI in the context of treating ADs, which is related to this thesis' investigation about the potential of RL in optimizing ET protocols.

4.2.1 AI-Assisted Anxiety Treatment

Medication and psychotherapy are two main treatment approaches for patients with ADs, while the combination of both is significantly more beneficial than medication alone [13]. To account for the lack of psychotherapy availability caused by limited resources [12], a group of researchers in China developed an AI psychotherapy robot including different treatment modules to help treat patients with ADs [129]. The authors propose a protocol that aims to determine whether medication plus AI-assisted psychotherapy has greater efficacy in the treatment of ADs than medication alone. They plan on recruiting 708 patients with ADs and randomly allocating them to either the medication plus AI-assisted psychotherapy group, or the medication alone group. Assessments based on different rating scales are planned to be conducted at baseline and at specified intervals with the overall goal of assessing the efficacy of medication plus AI-assisted psychotherapy for ADs.

Bâlan et al. developed a ML approach for automatic phobia therapy [10]. The authors introduced a Virtual Reality (VR) environment for treating acrophobia (fear of heights) which relies on gradual exposure to stimuli and accompanied measurements of physiological signals. They developed a VR-based game for which they trained and tested two classifiers: one determining the patient's current level of fear based on physiological measurements and one estimating the next scenario of exposure in the game based on the current level of fear. The game starts at the ground floor, while the classifiers determine the next level of exposure. Bâlan and colleagues plan to perform a series of experiments to evaluate the efficacy of the VR environment in treating acrophobia. They also introduced a novel approach towards using an intelligent virtual therapist recognizing human emotions based on biophysical signals, giving advice, providing encouragement and adapting the exposure scenario according to the subject's affective state. In a first experiment, five subjects repeatedly played the VR game with and without assistance from the virtual therapist, where statistically significant results could be achieved with the subjects finishing the game quicker and showing decreased fear based on physiological signals in the case of assistance from the virtual therapist.

An example for the application of RL in the treatment of phobias is the work by Mahmoudi-Nejad and colleagues [87], which propose a framework for treating arachnophobia (spider phobia) that adapts to individual subjects using RL. Mahmoudi-Nejad and colleagues worked with virtual patients implemented based on prior arachnophobia psychology research. In their framework, the RL agent acts as therapist, dynamically generating spiders with the goal of keeping the patient within a defined physiological state to allow for more effective ET. They use physiological responses of subjects when interacting with a virtual spider to estimate their real-time stress levels. The RL agent then modifies the virtual spider to reach a predefined target stress level. The virtual spiders have movement- and appearance-related attributes that are initially generated randomly and later adjusted by the RL approach. In evaluation simulations, Mahmoudi-Nejad and colleagues showed that their approach outperformed three tested baseline methods in the number of spiders shown to the virtual subjects before finding one inducing the subject's target stress level.

4.3 Cognitive Models Enhancing AI

In this thesis, cognitive models of associative learning are used as a stand-in for patients, with which an RL agent is interacting in order to extinguish a previously acquired fear. The approach of using cognitive models before empirical data collection is in line with the recently proposed "cognitive priors" approach, which has been motivated by the fact that AI algorithms generally require large amounts of data while available resources are often limited. The "cognitive priors" approach was proposed by Bourgin and colleagues [20] and comprises the idea of requiring less empirical data by using existing psychological models to generate synthetic behavioral data, on which an AI algorithm is pre-trained. Bourgin et al.'s work focuses on the prediction of human decisions, as they generated synthetic data from cognitive models of decision making, which was then incorporated as inductive bias into a neural network. In their work a neural network was trained on two synthetic datasets and then fine-tuned using a much smaller dataset of real human data. Bourgin and colleagues demonstrated superior performance over previously suggested cognitive and ML approaches for predicting human decisions. They also found that the resulting cognitive model priors allowed the network to achieve significantly better generalization performance with fewer training iterations than equivalent models with no such priors. With their "cognitive priors" approach, Bourgin et al. showed that cognitive models can establish prior theoretical knowledge used to bootstrap AI algorithms with the effect of reduced need to collect empirical data.

Trafton et al. used the "cognitive priors" approach in their recent research to predict human behavior from limited amounts of data [133]. They introduced a method for generating synthetic data of human behavior using cognitive models, which was then used to train a deep network to predict behavior based on outward observations. In more detail, they utilized a computational cognitive architecture to model the internal states and different strategies that people chose in a dynamic supervisory control task. The data generated by the cognitive model was used to train a deep network predicting human actions during the performance of the control task. Using the cognitive models to generate synthetic data, the authors could overcome the lack of available data while also introducing variability to achieve a balanced training set. The network trained on the synthetic data far outperformed the deep networks trained only on limited empirical data.

Also Schürmann and colleagues [120] considered the "cognitive priors" approach in their article on how they expect cognitive modeling to improve the understanding of individual cognitive processes in human-agent interaction. They differentiate between three applications of cognitive models: (1) using models of human agents to improve predictions of the agent's behavior (models of agents), (2) modeling human behavior to pre-train and adapt interaction (models of humans), and (3) generating behavior of an artificial agent based on a cognitive model of human behavior (models of interaction). Combining the idea of human and robot models with the "cognitive priors" approach, they argue that agents like humanoid robots could produce more human-like behavior while better adapting to the human partner.

In their recent work, Howes and colleagues [63] proposed an approach based on computational rationality, a model which brings together RL and cognitive modeling facilitating computational understanding of humans. They argue that latent states and processes in human cognition are not captured easily by model-free learning methods making them insufficient for cooperative AI. In order to build cooperative agents, it was not only required to estimate the state of a human partner, but also to predict how people adapt their behavior in response to an agent's actions. Howes et al. propose the usage of computationally rational models, which are information processing models of cognition, that make predictions based on a decision policy that is optimally adapted to internal factors limiting human behavior (e.g., individual capacities or past experiences). Computationally rational models can serve as a prior in probabilistic methods to help AI reason better about people, as they put psychological constructs at the center of RL algorithms, which enable understanding the contributions of a person's goals, capabilities, and environment to their behavior.

5 Methodology

In this thesis, RL is used to simulate therapy sessions where the RL agent functions as a stand-in for a therapist and computational associative learning models are used as a stand-in for patients. The aim is to obtain a clearer understanding of the limitations and possibilities of RL in optimizing ET protocols, with the long-term goal of optimized, personalized and widely accessible ET. This thesis' focus is on factors determining the success or failure of applying RL in this context, like the usage of different RL environmental settings, different patient model specifications and reward functions. This chapter outlines the methodology used for conducting and evaluating the simulations, where Section 5.1 gives as overview of the simulation design, Section 5.2 outlines mathematical details of the patient models used, Section 5.2.4 describes two different patient model observation functions, Section 5.3 defines the reward functions calculating the RL agent's feedback, Section 5.4 outlines specifications for the RL model hyperparameter tuning, and, lastly, Section 5.5 defines the protocol used for evaluating the trained RL models.

5.1 Simulation Design

As explained in Chapter 1, several procedural variables of ET are not concretely specified, leading to difficulties in the clinical application of ET [17]. To approach the lack of specification and provide optimized and personalized ET protocols, this thesis uses an RL framework, where the RL agent operates as therapist interacting with a patient in an RL environment. The steps in the RL environment include acquisition of a patient's fear, the attempt to extinguish it following the RL agent's chosen ET protocol and subsequently testing the success of the ET protocol. Those three steps coincide with the three stages of the classical FC paradigm (see Section 3.3.1), where EL (stage 2) is formalized as a (partially observable) Markov Decision Process (PO)MDP. The RL agent estimates the patient's state of fear through observing responses (CRs) and then chooses one of the available actions according to its current policy. The goal of the RL algorithm is to maximize fear extinction measured in the RoF test (stage 3) with the level of extinction serving as reward for the RL algorithm [89].

The RL algorithm used for training the models is Stable Baselines3's [111] RecurrentPPO with an MlpLstm policy (see Section 3.5.1). The default model parameters can be found in Stable Baselines3's documentation for the RecurrentPPO with MlpLstm policy [111]. Some of these parameters have been object of the hyperparameter tuning process which is outlined in Section 5.4.

For conducting the simulations, the classical RL problem (see Figure 3.3) has been

adapted to this thesis' problem setting (see Figure 5.1). As described above, the RL agent functions as a therapist trying to optimize an ET protocol through interacting with the RL environment. The environment consists of a patient, implemented by a computational associative learning model, passing through the three stages of the standard fear conditioning paradigm (see Figure 3.1).

Each action of the RL agent is referred to as a trial and corresponds to a stimuli presentation (certain combination of CSs and US; all available actions are described in Section 5.1.1). On a high level, one RL episode corresponds to one patient passing through the three stages and consists of the following steps (see also Figure 5.1):

- A patient, represented by the chosen patient model (RW, KRW or DP-KRW), gets instantiated with specified patient model parameters
- The patient runs through the Acquisition (ACQ) phase (stage 1), where a negative association for one CS (referred to as CS+) is acquired by repeatedly negatively reinforcing it, while another CS (referred to as CS-) is kept neutral by presenting it without negative reinforcement.
- After passing stage 1, the patient runs through the extinction phase (stage 2), which is where the RL agent chooses actions (stimuli presentations) with the goal of extinguishing the CS+'s negative association. After each trial, which corresponds to one RL training step, the RL agents observes the patient's CR which serves as a measure for the current level of fear. A reasonable choice for the number of trials n in stage 2 is subject of one of this thesis' SQs.
- After the n^{th} trial, the patient runs through a RoF test (stage 3), from which a reward is calculated based on the patient's CRs during stage 3, serving as (delayed) feedback for the RL agent on the success of the extinction protocol chosen in stage 2. Finding an appropriate reward function that leads to a successful learning is one of the SQs approached in this thesis.

For implementing the RL environment, *Gym* has been utilized, which provides an API to communicate between learning algorithms and environments [23]. Gym offers a diverse collection of environments while also providing the opportunity of implementing custom environments. Implementation details for the custom Gym environment used in this thesis are outlined in the following section.

5.1.1 RL Environmental Specifications

As mentioned before, in the experiments two CSs have been considered: CS+ and for reference CS-. Due to the possible absence/presence of the individual CSs and the US in each trial, the number of potential actions for the RL agent is $2^{\text{number of CSs}+1}$. For the two CSs used in the simulations, eight potential actions exist, which will be abbreviated by [presence CS+ (0/1), presence CS- (0/1), presence US (0/1)] hereafter. The action vector [1,0,1] would, e.g., indicate the presentation of CS+ with US, and therefore a negative reinforcement of CS+. Due to the absence of both CSs, actions [0,0,0] and [0,0,1] have

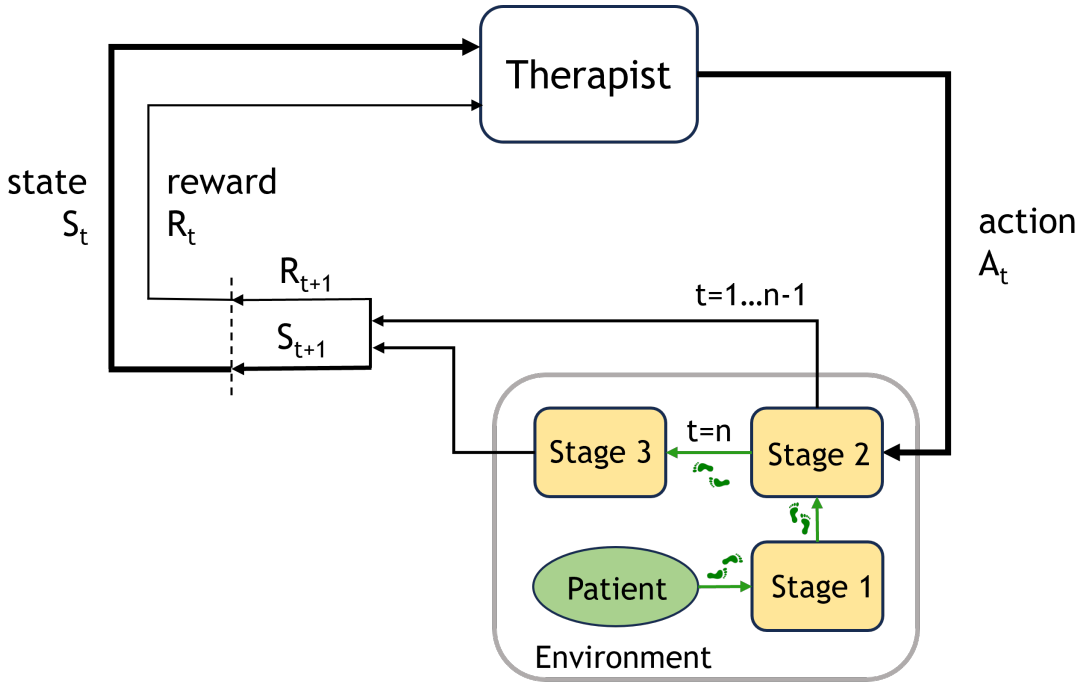


Figure 5.1: Classical RL problem adapted to this thesis' problem setting. The RL agent functions as a therapist with the goal of optimizing an ET protocol to extinct a patient's fear. The RL environment consists of a patient, implemented by a computational associative learning model, which runs through stages 1-3 of the standard FC paradigm (see Figure 3.1). The RL agent's influence is limited to stage 2, where each action corresponds to a trial at timestep t, i.e. a presentation of stimuli, constituting the fear extinction protocol. There is a total of n trials. The reward is set to 0 for the first $n-1$ trials. After the n^{th} trial, the patient runs trough stage 3 (RoF testing), from which a reward is calculated based on a specified reward function, giving the therapist feedback on the success of the ET protocol. After each trial the RL agent observes the patient's CR, which serves as a measure of the patient's current level of fear.

been excluded, leading to $8 - 2 = 6$ available actions for the RL agent. Using the Gym space *Discrete*, the resulting action space is Discrete($2^{\text{number of CSs+1}} - 2$)=Discrete(6), where the six discrete actions are [0,1,0], [0,1,1], [1,0,0], [1,0,1], [1,1,0], [1,1,1].

After each stimuli presentation during stage 2, the RL agent observes the patient's CR, which serves as a measure for the patient's current level of fear. The magnitude of the CR is dependent on the strength of the negative association towards the presented stimuli, which evolves according to the dynamics of the respective patient model (see Section 5.2). The observation space is implemented using the Gym space *Box* with a single scalar observation (shape=(1,)). Assuming a total of n trials, the RL agent's reward is set to 0 during whole stage 2. After the n^{th} trial the patient runs through stage 3 (RoF testing), from which a reward is calculated, giving the therapist delayed feedback on the success of the chosen ET protocol. Implementation details for the individual FC stages are discussed in the following.

Stage 1 - Fear Acquisition

Stage 1 consists of a total number of 64 trials with the goal of forming a negative association for CS+ (but not for the control stimulus CS-). Of the 64 trials, 32 trials consist of CS+ presentations with a negative reinforcement proportion of 0.75 and 32 trials consist of CS- presentations, which are never negatively reinforced. Stage 1's simulation setup looks like the following: 24 times [1,0,1], 8 times [1,0,0], 32 times [0,1,0] where the order is randomly permuted. The non-reinforced trials have been incorporated to be in line with typical lab setups allowing to isolate the effect of the US by contrasting [1,0,1] and [1,0,0] trials and preventing the habituation to the US [81]. While the number of trials in common FC experiments is around 5 to 20, for this thesis' simulations 32 CS+ presentations have been chosen to allow for effective FC even with low patient learning rates.

Stage 2 - Fear Extinction

During stage 2 the RL agent has a defined number of trials in order to extinct the fear acquired in stage 1. Being object of one of this thesis' SQs, different values for the number n of stage 2 trials have been tested (see Section 7.1). The RL agent observes the patient's CR after each trial. The reward during whole stage 2 is set to 0, while the RL agent's delayed feedback on the effectiveness of the chosen ET protocol is calculated based on the RoF test, see stage 3.

Stage 3 - Extinction Retention

Stage 3 serves as RoF test which evaluates in how far the negative association acquired in stage 1 could be extinguished during stage 2. In this thesis' simulation setup, stage 3 consists of 5 alternating presentations of CS+ and CS- without US, which is in line with the general design principle for FC experiments of only using a small number of trials in the RoF test [81]. Providing insight into the successful extinction of fear during stage 2,

the CRs during stage 3 are used for calculating the reward. Different reward functions have been tested and are outlined in Section 5.3.

5.2 Patient Models

In this thesis' simulations three different computational associative learning models have been used as a stand-in for the patients. While the general concepts have been outlined in Section 3.4, this section focuses on mathematical details of the individual models.

5.2.1 Rescorla-Wagner Model

A CS can consist of several component stimuli (compound CS), where the change of the associative strength of each component stimulus is dependent on the associative strength associated with the whole compound CS (aggregate associative strength). Assuming that a trial n is described by a vector $x_n = (x_1(n), x_2(n), ..., x_d(n))^T$ where $x_i(n) = 1$ if CS_i is presented and $x_i(n) = 0$ otherwise. The RW model deals with changes of associative strengths from trial to trial not considering details within or between trials and is therefore a trial-level model. The aggregate associative strength for trial n can be thought of as the US prediction and is

$$v_n = w_n^T x_n \tag{5.1}$$

with the d-dimensional vector of associative strengths w. The update of the associative strength vector w_n to w_{n+1} is conducted as

$$w_{n+1} = w_n + \alpha \delta_n x_n \tag{5.2}$$

with the learning rate $\alpha \in [0, 1]$ and the prediction error

$$\delta_n = r_n - v_n. \tag{5.3}$$

r_n is the magnitude of the US on a trial n and therefore the target of the prediction. Due to the factor x_n in Equation (5.2), the results of a trial only includes the associative strengths of CS components present on that trial. The aggregate associative strength can be thought of as expectation of the US magnitude and the prediction error as a measure of surprise, describing the deviation of the expected US magnitude from the actual one [48]. The RW model therefore formalizes two important principles: (1) learning is driven by reward prediction errors and (2) simultaneously presented stimuli summate to predict reward.

The RW model parameters specified for the simulations are the initial weights w_0 and the learning rate α. Specific choices for these parameters are outlined in Table 6.1.

5.2.2 Kalman Rescorla-Wagner Model

For a probabilistic interpretation of the RW model, a set of probabilistic assumptions has to be made about how the patient generates sensory data. This internal mode of the patient

is defined by a prior on weights, $p(w_0)$, a change process on the weights, $p(w_n|w_{n-1})$, as well as a reward distribution given stimuli x_n and weights w_n, $p(r_n|w_n, x_n)$. Following earlier work [48, 78] these distributions can be assumed to characterize a linear-Gaussian dynamical system (LDS):

$$w_0 \sim \mathcal{N}(0, \sigma_w^2 I) \tag{5.4}$$

$$w_n \sim \mathcal{N}(w_{n-1}, \tau^2 I) \tag{5.5}$$

$$r_n \sim \mathcal{N}(v_n, \sigma_r^2) \tag{5.6}$$

with the identity matrix I. The intuitive claims of this LDS about the patient's internal mode are: (1) the weights tend to be close to 0 (weak associations) and the strength of the prior on weights is inversely proportional to the variance of the initial weight diffusion σ_w^2, (2) weights tend to change slowly and independently over time with the volatility of this change process increasing with the variance of the state diffusion τ^2, and (3) reward is a linear combination of stimulus activations, with additive noise [48]. Being an instantiation of the least mean squares (Widrow-Hoff) learning rule [138], the RW model has a normative basis in statistical estimation, but with only maintaining a single point hypothesis about the weights and ignoring uncertainty, it is not a fully probabilistic estimator. Research has found that the brain maintains representations of uncertainty [6] while updating these representations through Bayesian inference [107]. The KRW model is a Bayesian model of learning, which proposes that learners represent uncertainty by formulating a posterior distribution over hypotheses based on the available data. For associative learning, Bayes' rule specifies the posterior distribution as follows:

$$p(w_n|x_{1:n}) \propto p(x_{1:n}|w_n)p(w_n) \tag{5.7}$$

where index $1:n$ denotes all trials from 1 to n (see also Section 3.4.2). The posterior, under the LDS specified in Equations (5.4)-(5.6), is Gaussian with mean w_n and covariance matrix $\sum_n$, which are updated using the following Kalman filter equations:

$$w_{n+1} = w_n + K_n \delta_n \tag{5.8}$$

$$\Sigma_{n+1} = \Sigma_n + \tau^2 I - K_n \left(x_n^T (\Sigma_n + \tau^2 I) \right) \tag{5.9}$$

with prediction error δ_n (see Equation (5.13)), $w_0 = 0$, $\sum_0 = \sigma_w^2 I$ and the *Kalman gain* K_n

$$K_n = \frac{(\Sigma_n + \tau^2 I) x_n}{x_n^T (\Sigma_n + \tau^2 I) x_n + \sigma_r^2}, \tag{5.10}$$

which is used analogously to the learning rate α in the RW model with variance of the observation noise σ_r^2. The Kalman gain is stimulus-specific, dynamic and grows monotonically with the uncertainty encoded in the diagonals of the posterior covariance matrix Σ_n. The second term of the covariance matrix update, expressed by $\tau^2 I$ in Equation (5.9), implies that uncertainty grows over time due to the random diffusion of the weights (Equation (5.5)). The growth of uncertainty over time is proportional to the diffusion variance τ^2, which leads to higher learning rates in more volatile environments.

The third term of the covariance matrix update, expressed by $K_n \left(x_n^T(\Sigma_n + \tau^2 I)\right)$ implies the reduction of uncertainty due to the observation of data. Therefore, the variance on the diagonal of the covariance matrix as well as off-diagonals for any correlated CSs are reduced, whenever a CS is observed. [48]

The KRW model parameters specified for the simulations are the initial associative weights w_0, the log-variance of the initial weight distribution $\log(\sigma_w^2)$, the log-variance of state diffusion $\log(\tau^2)$ and the log-variance of the observation noise $\log(\sigma_r^2)$. Specific choices for these parameters are outlined in Table 6.2.

5.2.3 Dirichlet Process-Kalman Rescorla-Wagner

The Dirichlet Process-Kalman Rescorla-Wagner (DP-KRW) model is an extension of the KRW model by considering the concept of latent causes (see Sec 3.4.3). The model implementation used in this thesis' simulations is based on the 2017 work of Gershman and colleagues [50], while their model has been adapted to the usage of the KRW model for this thesis' simulations. Due to close similarities between KRW and DP-KRW, some details regarding individual parameters are omitted in the following, but can be found in the description of the KRW model (see Section 5.2.2).

The update equations of the DP-KRW model have additionally to the timestep n a dimension for the latent cause k. The update equation for the weights $w_{k,n+1}$ and the covariance $\Sigma_{k,n+1}$ in the DP-KRW are given by

$$w_{k,n+1} = w_{k,n} + K_{k,n}\delta_n \tag{5.11}$$

$$\Sigma_{k,n+1} = \Sigma_{k,n} + \tau^2 I - p_{Z_{k,n+1}} \left(K_{k,n} \left(x_n^T(\Sigma_{k,n} + \tau^2 I)\right)\right) \tag{5.12}$$

with initial weights $w_{k,0} = 0$ and initial covariance matrix $\Sigma_{k,0} = \sigma_w^2 I$ for all latent causes $k = 1, ...k_{\max}$. The scalar prediction error δ_n is calculated by

$$\delta_n = r_n - (v_n^T p_{Z_{n+1}}) \tag{5.13}$$

where $p_{Z_{n+1}}$ is the $k_{\max}$-dimensional posterior distribution of all latent causes (will be discussed later) and r_n represents the magnitude of the US on trial n. v_n is the $k_{\max}$-dimensional aggregate associative strength (expected US) calculated by

$$v_n = W_n x_n \tag{5.14}$$

where W_n is the matrix of weights for all latent causes with dimension $k_{\max}\times$number of CSs. The Kalman gain $K_{k,n}$ is defined as

$$K_{k,n} = \frac{p_{Z_{k,n+1}}(\Sigma_{k,n} + \tau^2 I)x_n}{p_{Z_{k,n+1}}^2 S_{k,n}}, \tag{5.15}$$

with the residual variance $S_{k,n}$ calculated by

$$S_{k,n} = x_n^T(\Sigma_{k,n} + \tau^2 I)x_n + \sigma_r^2. \tag{5.16}$$

The $k_{\max}$-dimensional posterior distribution $p_{Z_{n+1}}$ of all latent causes is calculated as

$$p_{Z_{n+1}} = \frac{e^{\log(p_{Z_n})+\boldsymbol{\ell}_n}}{\sum e^{\log(p_{Z_n})+\boldsymbol{\ell}_n}}, \tag{5.17}$$

with prior p_{Z_n} and log-likelihood $\boldsymbol{\ell}_n$ of all latent causes. The calculation of the $k_{\max}$-dimensional prior p_{Z_n} for the latent causes is determined by a Chinese restaurant process [45], with concentration parameter α_{DP} defining the probability of a new latent cause and stickiness parameter β increasing the probability of the latent cause from the previous trial:

$$p_{Z_{k,n+1}} = \begin{cases} \alpha_{\text{DP}} & \text{if } k = g_n + 1 \text{ (new latent cause)} \\ p_{Z_{k,n}} + \beta & \text{if } k = h_n \text{ (currently active latent cause)} \\ p_{Z_{k,n}} & \text{otherwise} \end{cases} \tag{5.18}$$

where g_n is the number of latent causes that have been active so far and h_n is the index of the currently active latent cause. The initial probability for latent cause 1 is 1 ($p_{Z_{1,0}} = 1$) and the $k_{\max}$-dimensional prior for all latent causes p_{Z_n} is normalized by $p_{Z_n} = \frac{p_{Z_n}}{\sum p_{Z_n}}$. The log-likelihood $\boldsymbol{\ell}_n$ for all latent causes follows a Gaussian distribution and is calculated by

$$\boldsymbol{\ell}_n = -\frac{1}{2}\left(\frac{r_n - W_n x_n}{\sqrt{S_n}}\right)^2 - \log(\sqrt{2\pi}\sqrt{S_n}) \tag{5.19}$$

where S_n is the residual variance of all latent causes. The updated active latent cause L_{n+1} is determined by a maximum a posteriori (MAP) approximation

$$L_{n+1} = \operatorname{argmax} p_{Z_{n+1}}. \tag{5.20}$$

The DP-KRW model parameters specified for the simulations are the initial associative weights w_0, the log-variance of the initial weight distribution $\log(\sigma_w^2)$, the log-variance of state diffusion $\log(\tau^2)$, the log-variance of the observation noise $\log(\sigma_r^2)$ (for further description see Section 5.2.2), the concentration parameter α_{DP}, the stickiness parameter β and the upper bound on the number of latent causes $k_{\max}$. Specific choices for these parameters are outlined in Table 6.3.

5.2.4 Patient Model Observation Functions

The patient model observation function is used to describe the relation between expected US and CR. In this thesis, two different observation functions have been used: a linear and a sigmoid one, representing two different mappings from expected US to emitted CR.

The respective functions used for the linear and the sigmoid observation function are outlined in the following two subsections.

Linear Observation Function

The calculation of the CR is dependent on the patient model and is defined below for using the linear observation function.

For the RW and the KRW model the CR of trial n is defined as

$$\mathrm{CR}_n = s(w_n^T x_n) + d \tag{5.21}$$

where s represents a 1-dimensional slope and d is an optional intercept.

For the DP-KRW model the CRs are calculated by

$$\mathrm{CR}_n = s((W_n x_n)^T p_{Z_{n+1}}) + d \tag{5.22}$$

with the posterior probabilities of the latent causes $p_{Z_{n+1}}$, the 1-dimensional slope s and the optional intercept d while all simulations have been conducted with $s = 1$ and $d = 0$. W_n is the matrix of weights for all latent causes with dimension $k_{\max} \times$ number of CSs.

One shortcoming of the linear observation function is that it does not constrain the CRs to be positive, and the negativity of the CRs grows linearly with the weights' negativity. Depending on the reward function used, these negative weights (e.g., for CS-) can be used by the RL agent to achieve high rewards while not effectively lowering the fear towards CS+ (which will become clearer once the reward functions get introduced in Section 5.3). With the linear observation function there is no diminishing utility that makes reducing a particular weight only profitable up to a point. A possible workaround is to use absolute values for the weights in the calculation of the CRs. In this case Equations (5.21)-(5.22) remain as specified above, except for substituting w with $|w|$. Another solution is using a differently shaped observation function, with a non-linear dependency of the CR on the weights. One example of such an observation function is the sigmoid function, which is outlined in the next section.

Sigmoid Observation Function

Using the sigmoid observation function leads to a close analogy between the RW model and the logisitic regression model [70]. Following Equation (5.13), for the RW and the KRW model the prediction error is calculated as follows

$$\begin{aligned} \delta_n &= r_n - \sigma(w_n^T x_n) \\ &= r_n - \frac{1}{1 + e^{-w_n^T x_n}}, \end{aligned} \tag{5.23}$$

and the emitted CR

$$\begin{aligned} \mathrm{CR}_n &= \sigma(w_n^T x_n) \\ &= \frac{1}{1 + e^{-w_n^T x_n}}. \end{aligned} \tag{5.24}$$

In the DP-KRW model the posterior probabilities $p_{Z_{n+1}}$ of the latent causes are considered in the calculation of the prediction error

$$\begin{aligned}\delta_n &= r_n - \sigma((W_n x_n)^T p_{Z_{n+1}}) \\ &= r_n - \frac{1}{1 + e^{-(W_n x_n)^T p_{Z_{n+1}}}}\end{aligned} \tag{5.25}$$

with the following calculation for the emitted CRs

$$\mathrm{CR}_n = \sigma((W_n x_n)^T p_{Z_{n+1}}) \tag{5.26}$$

where again W_n is the matrix of weights for all latent causes with dimension $k_{\max} \times$ number of CSs. It has to be noted that with the sigmoid observation function the patient models' parameter settings of $w_0 = 0$ lead to a CR of 0.5 in the beginning of stage 1 translating into a certain level of initial fear, while in the case of the linear observation function the simulations start with CRs of 0.

5.3 Reward Functions

A crucial parameter for the success of RL simulations is the choice on the reward function. The RL agent's goal is to maximize the reward and – by doing so – to reach a defined goal, like e.g., lowering a patient's CR. In the course of this thesis, six different reward functions have been tested. While some focus on the CRs to the CS+, others also take the CRs to the CS- into account, some focus on the CRs during Retention (RET) and others put it into perspective by also considering the CRs during Acquisition (ACQ). The choice of functions tested has been inspired by the collection of reward functions reported by Lonsdorf et al. [83]. The mathematical details of the individual reward functions are outlined in the following.

The first reward function, referred to as RETbasedReward, considers the CRs to all CS+ presentations during stage 3 (RET), providing a measure for the level of negative association towards CS+ during RET

$$R = -\frac{1}{p} \sum_{i=1}^{p} CR^{+}_{RET,i} \tag{5.27}$$

where p is the number of CS+ presentations in the RoF test, with $p = 5$ in this thesis' experiments.

The next reward function, SummedRETbasedReward, considers the CRs to all CS+ and CS- presentations during stage 3, providing a measure for the level of negative association towards CS+ and CS- during RET

$$R = -\frac{1}{p+q} \left(\sum_{i=1}^{p} CR^{+}_{RET,i} + \sum_{j=1}^{q} CR^{-}_{RET,j} \right) \tag{5.28}$$

where p and q are the number of CS+ and CS- presentations in the RoF test respectively, with $p, q = 5$ in this thesis' experiments.

Another reward function tested, referred to as absSummedRETbasedReward, is similar to SummedRETbasedReward (see Equation (5.28)) with the difference of using absolute CRs, indicated by absolute value bars $|\cdot|$ in Equation (5.29)

$$R = -\frac{1}{p+q}\left(\sum_{i=1}^{p}|CR^{+}_{RET,i}| + \sum_{j=1}^{q}|CR^{-}_{RET,j}|\right). \tag{5.29}$$

Using absolute values has been considered to counteract the RL agent's behaviour of aiming at negative CRs in order to maximize the reward calculated by SummedRETbasedReward (as mentioned in Section 5.2.4).

When rating the success of fear extinction, instead of solely focusing on the CRs during the RoF test, also the CRs at the end of stage 1 can be taken into account. By doing so, the focus is not only on the magnitude of fear after extinction, but rather on how much a fear could be decreased compared to the magnitude of fear before the extinction phase. By applying a scaling, the return value is a percentage of fear extinction, with 0 indicating no fear extinguished and 100 indicating all fear extinguished. The following reward function is referred to as ACQcorrectedNonDifferentialReward and focuses on the CRs of the m and k first/last CS+ during RET/ACQ respectively. It is inspired by Lonsdorf et al.'s work from 2019 [83, Table 1, index 2 and 6]

$$R = 100 - \left(100 * \frac{\frac{1}{m}\sum_{i=1}^{m} CR^{+}_{RET,i}}{\frac{1}{k}\sum_{j=n-k+1}^{n} CR^{+}_{ACQ,j})}\right) \tag{5.30}$$

where n is the total number of ACQ trials ($n = 64$) and with the choice of $m, k = 3$ in this thesis' experiments.

The next reward function, referred to as ACQcorrectedDifferentialReward, is also offering a correction for ACQ but while Equation (5.30) only considers the CRs to the CS+, this reward function also considers the CRs to the CS-. This is done by subtracting the mean of the CRs to the first m CS+ and the CRs to the first m CS- presentations during RET and dividing it by the subtraction of the CRs to the last k CS+ and the CRs to the last k CS- presentations during ACQ. Considering the CRs not only during RET, but also during ACQ has the same purpose as in ACQcorrectedNonDifferentialReward, namely to establish a relation between the fear level after ACQ and the fear level after the extinction phase and therefore to express how much a fear could be decreased compared to the magnitude of fear before the extinction phase. Considering the CRs not only to CS+ but also to CS- puts the CRs to CS+ in a relation to possibly non-zero CRs to CS-, which might be the case due to specified initial weights w_0 or due to the ET protocol chosen by the RL agent. If initial weights w_0 are set in a way, that from the start of ACQ

on a negative association towards both CSs is present, this can be corrected for through considering the CRs to both CS+ and CS- when assessing the success of the RL agents ET protocol. ACQcorrectedDifferentialReward is also inspired by Lonsdorf et al.'s work [83, Table 1, index 10 and 13]

$$R = 100 - \left(100 * \frac{\frac{1}{m}(\sum_{i=1}^{m} CR^{+}_{RET,i} - \sum_{i=1}^{m} CR^{-}_{RET,i})}{\frac{1}{k}(\sum_{j=n-k+1}^{n} CR^{+}_{ACQ,j} - \sum_{j=n-k+1}^{n} CR^{-}_{ACQ,j})} \right) \tag{5.31}$$

where n is the total number of ACQ trials ($n = 64$) and with the choice of $m, k = 3$ in this thesis' experiments. The return value is a percentage with 0 indicating no fear extinguished and 100 indicating all fear extinguished.

The next reward function, referred to as ACQcorrectedNormReward, is inspired by ACQcorrectedDifferentialReward (see Equation (5.31)), but instead of subtracting the means of the CRs to CS+ and CS-, the euclidean norm (denoted by $\|\cdot\|$ in Equation (5.32)) is calculated from an array consisting of the mean CR to the first m CS+ and the mean CR to the first m CS- during RET for the numerator and from an array consisting of the mean CR to the last k CS+ and the mean CR to the last k CS- during ACQ for the denominator

$$R = 100 - \left(100 * \frac{\|(\frac{1}{m}\sum_{i=1}^{m} CR^{+}_{RET,i}\ , \frac{1}{m}\sum_{i=1}^{m} CR^{-}_{RET,i})\|}{\|(\frac{1}{k}\sum_{j=n-k+1}^{n} CR^{+}_{ACQ,j}\ , \frac{1}{k}\sum_{j=n-k+1}^{n} CR^{-}_{ACQ,j})\|} \right) \tag{5.32}$$

where n is the total number of ACQ trials ($n = 64$) and with the choice of $m, k = 3$ in this thesis' experiments. The intuition behind using ACQcorrectedNormReward is the same as in ACQcorrectedDifferentialReward, namely to put the fear level after in RET in a relation to the fear level after ACQ and to correct for non-zero CRs to CS-.

5.4 Hyperparameter Tuning

For hyperparameter tuning, RL Baselines3 Zoo [110] has been used, which utilizes Optuna [3], an advanced hyperparameter optimization framework enabling users to adopt state-of-the-art algorithms for tuning hyperparameters.

An Optuna study runs a user defined number of Optuna trials iteratively with the aim of finding the hyperparameter combination that best optimizes the objective function. A trial is the process of evaluating an objective function using a set of hyperparameters and an Optuna study corresponds to an optimization task, i.e., a set of trials. Optuna study settings of this thesis' hyperparameter tuning are outlined in Section 6.2.

Table 5.1 outlines parameters of the RecurrentPPO with MlpLstm policy that are considered in the RL Baselines3 Zoo hyperparameter tuning and a short explanation on an intuitive level is offered.

Table 5.1: Overview of parameters for RecurrentPPO model with MlpLstm policy that have been considered for hyperparameter tuning [111].

Hyperparameter	**Description**
batch_size	minibatch size; corresponds to how many experiences are used for each gradient descent update, where batch size is inversely proportional to the training time
n_steps	number of steps to run before performing updates on the policy and value function; large values may introduce more bias into the estimation while small values may result in higher variance and slower convergence
gamma	discount factor adjusting the importance of rewards over time; higher values mean more future rewards are taken into account when determining the optimal action choices (corresponds to γ in Equation (3.6))
learning_rate	corresponds to the strength of each gradient descent update step; too high values can cause too large updates with unstable optimization processes, while too small values can cause too conservative updates with a slowed down optimization process
ent_coef	entropy coefficient for the loss calculation; higher values encourage more exploration, while lower values potentially lead to a more deterministic policy (corresponds to c_2 in Equation (3.25))
clip_range	clipping parameter, determining the threshold within which policy updates are restricted and therefore controlling the stability of the training (corresponds to ϵ in Equation (3.17))
n_epochs	number of passes through the experience buffer during gradient descent; larger batch sizes lead to larger acceptable values for n_epochs, lower values will ensure more stable updates, at the cost of slower learning
gae_lambda	Generalized Advantage Estimation (GAE) parameter, used to control the trade-off between bias and variance in estimating the advantages during the training process
max_grad_norm	maximum value for the gradient clipping regularization technique, used to prevent exploding gradients by ensuring that the gradients remain within a reasonable range and to help stabilize the training process
vf_coef	value function coefficient for the loss calculation, controls the importance of the value function loss relative to the policy loss during optimization (corresponds to c_1 in Equation (3.25))
net_arch	specification of the policy and value networks, defining the number of layers and the number of neurons in each layer of the MLP part of the MlpLstm architecture.
activation_function	activation function applied to the output of the nodes in the MlpLstm architecture to introduce non-linearity and capture complex patterns in the data

The hyperparameter ranges for the tuning process are taken from RL Baselines3 Zoo [110] and specified in Table 5.2.

Table 5.2: Hyperparameter ranges for Optuna hyperparameter tuning.

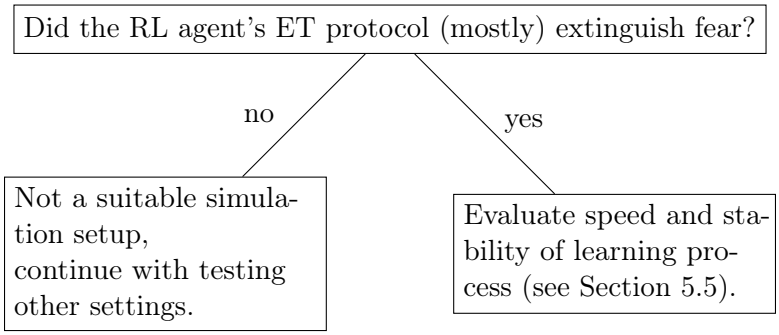
Parameter	**Range**
batch_size	∈ [8, 16, 32, 64, 128, 256, 512]
n_steps	∈ [8, 16, 32, 64, 128, 256, 512, 1024, 2048]
gamma	∈ [0.9, 0.95, 0.98, 0.99, 0.995, 0.999, 0.9999]
learning_rate	∈ [1e-5, 1]
ent_coef	∈ [1e-8, 0.1]
clip_range	∈ [0.1, 0.2, 0.3, 0.4]
n_epochs	∈ [1, 5, 10, 20]
gae_lambda	∈ [0.8, 0.9, 0.92, 0.95, 0.98, 0.99, 1.0]
max_grad_norm	∈ $[0.3, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 2, 5]$
vf_coef	∈ [0, 1]
net_arch	∈ [small, medium]
activation_function	∈ [tanh, relu]

The parameter *net_arch* includes specifications of the policy and value networks. *Small* represents dict(pi = $[64, 64]$, vf = $[64, 64]$), which specifies a two-layer neural network with 64 neurons in each layer for the policy network (pi) as well as for the value function network (vf). *Medium* stands for dict(pi = $[256, 256]$, vf = $[256, 256]$), which specifies a two-layer neural network with 256 neurons in each layer for the policy network as well as for the value function network.

For the evaluation of hyperparameter importances Optuna's default importance evaluator FanovaImportanceEvaluator [65] has been used, which utilizes an empirical performance model based on random forests in order to analyze how much of the performance variance in the configuration space is explained by single parameters.

5.5 Evaluation of RL Policies

In order for an RL policy to be considered successful, it has to be able to extinguish the patient's previously acquired negative associations and the learning process has to be comparatively fast and stable. Therefore, the trained models have been evaluated based on the following two questions: (1) Has the patient's fear been (mostly) extinguished with the RL agent's proposed ET protocol? and if yes, (2) how fast and stable was the learning process? This 2-step evaluation process is schematically presented in the following decision tree:



For the RW model the most efficient and therefore aimed-for extinction protocol is the repeated presentation of a CS in the absence of the US with which it was previously paired [60, 103]. While the optimal extinction protocol might slightly differ for the KRW and the DP-KRW model, the simulations conducted with the RW model have been evaluated based on whether the ET protocol with only CS+-alone presentations, referred to as [1,0,0]-protocol, has been found. To analyze whether this was the case, two aspects have been considered: (1) How many of all possible actions during evaluation have been CS+-alone presentations ([1,0,0]-fraction) and (2) in how many evaluation episodes *all* extinction trials have been CS+-alone presentations (all-[1,0,0]-fraction)? A definition of the [1,0,0]-fraction and the all-[1,0,0]-fraction is outlined in the following.

[1,0,0]-fraction

To have a measure for how many of all possible actions during evaluation have been CS+-alone presentations, the number of chosen [1,0,0]-actions during evaluations has been divided by the total number of total evaluation actions

$$\text{[1,0,0]-fraction} = \frac{\sum_{i=1}^{s_{\text{tot}}} n_{[1,0,0],i}}{s_{\text{tot}} e_{\text{tot}} t_{\text{tot}}} \tag{5.33}$$

where $n_{[1,0,0],i}$ is the number of [1,0,0]-actions across all episodes and trials of the i^{th} simulation, s_{tot} is the total number of simulations, e_{tot} is the number of episodes used for evaluation per simulation and t_{tot} is the number of trials per episode. A higher value for [1,0,0]-fraction means more [1,0,0]-actions found and is therefore favorable.

All-[1,0,0]-fraction

To analyze for how many evaluation episodes the model suggests all actions to be CS+-alone presentations, the number of all-[1,0,0] episodes has been divided by the number of total evaluation episodes

$$\text{all-[1,0,0]-fraction} = \frac{\sum_{i}^{s_{\text{tot}}} \mathbf{1}(\text{all-[1,0,0]})_i}{s_{\text{tot}} e_{\text{tot}}} \tag{5.34}$$

where again s_{tot} represents the total number of simulations, e_{tot} the number of episodes used for evaluation per simulation and the indicator function $\mathbf{1}$(all-[1,0,0]) being defined

as

$$\mathbf{1}(\text{all-[1,0,0]}) = \begin{cases} 1, \text{ if all episode actions are [1, 0, 0]} \\ 0, \text{ otherwise} \end{cases} \tag{5.35}$$

A higher all-[1,0,0]-fraction means that in more episodes all actions have been [1,0,0] and is therefore favorable.

Analysis of Learning Curves

The second evaluation step consists of analyzing the learning curves of the different simulation settings. Mean learning curves have been calculated from simulations using the exact same simulation setup (i.e., same patient model, patient model observation function, reward function and hyperparameters). The formula for the mean learning curve is

$$\text{mean\_LC}(j) = \frac{1}{n_{\text{sim}}} \sum_{i=1}^{n_{\text{sim}}} \frac{1}{n_{\text{ep}}} \sum_{k=1}^{n_{\text{ep}}} R_{i,j,k}, \quad j \in [1, n_{\text{eval}}] \tag{5.36}$$

where n_{sim} is the number of simulations per setting, n_{eval} is the number of evaluation callbacks per simulation, n_{ep} the number of episodes per evaluation callback, i represents the index of the simulation, j the index of the evaluation callback and k the index of the evaluation episode. $R_{i,j,k}$ therefore represents the reward of the k^{th} evaluation episode of the j^{th} evaluation callback in the i^{th} simulation.

In a further step, non-linear least squares has been used to fit the following increasing form of the exponential decay function to the average learning curves

$$y(t) = R_{\text{lim}} - (R_{\text{lim}} - R_0)e^{(-t/\tau)} \tag{5.37}$$

where R_{lim} is the limiting value which can be interpreted as the approached maximum reward with the learned extinction policy, R_0 is the mean reward of the initial evaluation callback and τ the characteristic time constant. Inserting $t = 5\tau$ gives

$$y(t) = R_{\text{lim}} - (R_{\text{lim}} - R_0) * e^{-5} \approx 0.9933 R_{\text{lim}} \tag{5.38}$$

showing that 5τ leads to $\sim 99.3\%$ of asymptotic performance and therefore the smaller τ, the faster the RL agent reaches R_{lim}. For the assessment of different mean learning curves, τ values are compared, favoring smaller values. To check the goodness of fit, i.e. how well the mean learning curve is described by the fitted curve, the Coefficient of determination (R^2) has been chosen as a measure.

During the simulations, each episode consists of one patient running through the FC stages where the number of training steps is given by the number of extinction trials (one training step = one extinction trial), and therefore the number of patients required for reaching the $\sim 99.3\%$ of asymptotic performance is calculated by

$$\text{patients required} = \frac{5\tau}{\text{extinction trials}}. \tag{5.39}$$

6 Experimental Setup

This chapter outlines choices for patient model parameters and further experimental details on training, evaluation and hyperparameter tuning.

6.1 Patient Model Parameters

An explanation of the parameters used in the different patient models has been given in Section 5.2. The following subsections outline specific choices for those parameters, which have been inspired by related work or which' choice is object of one of this thesis' SQs (see Chapter 2).

RW Model Parameters

The RW patient model parameters are the initial weights w_0, and the learning rate α, which have been described in more detail in Section 5.2.1. While w_0 has been set to 0, an appropriate prior for the learning rate α is object of investigation and covered in Section 7.1. The specific parameter choices are summarized in Table 6.1.

Table 6.1: Overview parameter choices for the RW model.

Parameter	**Description**	**Value**
α	learning rate	see Section 7.1
w_0	initial associative weights	0

KRW Model Parameters

The KRW patient model parameters are the initial weights w_0, the log-variance of the initial weight distribution $\log(\sigma_w^2)$, the log-variance of the state diffusion $\log(\tau^2)$ and the log-variance of the observation noise $\log(\sigma_r^2)$, which have been described in more detail in Section 5.2.2. The simulations have been conducted using $w_0 = 0$, $\log(\sigma_w^2) = 0$, and $\log(\sigma_r^2) = 1.3863$ (values taken from Kruschke [78]). The parameter $\log(\tau^2)$ has been object of further investigations (see Section 7.3.1). The specific parameter choices are summarized in Table 6.2.

Table 6.2: Overview parameter choices for the KRW patient model.

Parameter	Description	Value
w_0	initial associative weights	0
$\log(\sigma_w^2)$	log-variance of the initial weight distribution	0
$\log(\tau^2)$	log-variance of state diffusion	see Section 7.3.1
$\log(\sigma_r^2)$	log-variance of the observation noise	1.3863

DP-KRW Model Parameters

The DP-KRW patient model parameters are the initial weights w_0, the log-variance of the initial weight distribution $\log(\sigma_w^2)$, the log-variance of the state diffusion $\log(\tau^2)$ and the log-variance of the observation noise $\log(\sigma_r^2)$, the concentration parameter α_{DP} and the stickiness parameter β, which have been described in more detail in Section 5.2.3. The simulations have been conducted using $w_0 = 0$, $log(\sigma_w^2) = 0$, $\log(\sigma_r^2) = 1.3863$ (values taken from Kruschke [78]), $\alpha_{\text{DP}} = 0.1$, $\beta = 0$ and $k_{\max} = 10$ (decision on values after discussion with Samuel Gershman, the author of [48]). Due to the small value for α_{DP}, small numbers of modes have higher probability, which more likely leads to the modification of existing modes than the creation of new ones. The parameter $\log(\tau^2)$ has been object of further investigations (see Section 7.3.1). The specific parameter choices are summarized in Table 6.3.

Table 6.3: Overview parameter choices for the DP-KRW patient model.

Parameter	Description	Value
w_0	initial associative weights	0
$\log(\sigma_w^2)$	log-variance of the initial weight distribution	0
$\log(\tau^2)$	log-variance of state diffusion	see Section 7.3.1
$\log(\sigma_r^2)$	log-variance of the observation (reward) noise	1.3863
α_{DP}	concentration parameter	0.1
β	stickiness of last mode	0
k_{max}	upper bound on number of modes	10

6.2 Further Experimental Details

This section outlines experimental details regarding the simulations, evaluations and hyperparameter tuning.

Simulation Details

Per simulation setting (i.e., a specific choice for the patient model, the patient model observation function and parameters, the reward function, and hyperparameters) 20

simulations have been conducted, using 100000 training steps each. In every simulation, 100 evaluation callbacks have been conducted (every 1000 of the 100000 timesteps), each consisting of 100 evaluation episodes with deterministic actions, i.e. always choosing the action with the highest probability. The considered learning curves for each simulation are constituted by the mean rewards of each evaluation callback. To get one learning curve per simulation setup from all 20 simulations, a mean training curve has been calculated according to Equation (5.36).

Evaluation Details

After training, each RL policy has been evaluated based on 1000 episodes with deterministic actions. With 20 simulations per specific simulation setup this leads to $1000 \cdot 20 = 20000$ evaluation episodes per simulations setup. The evaluation episodes have been used to analyze the chosen ET protocols by looking at the chosen stimuli presentations (actions).

Hyperparameter Tuning Details

The Optuna study for this thesis' hyperparameter tuning involved 500 trials using Optuna's default TPESampler (see Section 3.6.1) and default MedianPruner, which prunes if a trial's best intermediate result is worse than the median of intermediate results of previous trials at the same step (see [3] for more details and Section 5.4 for definition of Optuna study and trial). All of the 500 trials consisted of 50000 training steps, and the model was evaluated twice during the training process (after 25000 and 50000 steps) with five environments for 100 episodes each. The choice on these settings has been inspired by examples in the RL Baselines3 Zoo documentation [110] and subsequently has been refined by trial and error.

7 Experiments and Results

In order to assess the limits and potential of RL in the optimization of ET protocols, this thesis makes use of an RL framework, where the RL agent (therapist) interacts with a computational associative learning model (patient), with the goal of finding an optimized ET protocol that maximizes fear extinction. This thesis addresses different SQs, which have been outlined in Chapter 2 and can be summarized as follows:

- SQ1: How many extinction trials lead to robust and efficient ET protocols in the simulations?
- SQ2: Which prior for the RW model learning rate captures empirically reasonable speed of human fear learning and supports finding optimized ET protocols in the simulations?
- SQ3: Which factors to consider in the reward function to help finding an optimized ET protocol?
- SQ4: How does the patient model observation function influence the RL agent's ability to find an optimized ET protocol?
- SQ5: Can parameter settings that lead to an optimized ET protocol for one patient model achieve similarly good results when transferred to other related patient models?

In order to answer these SQs, several simulations have been conducted using the mentioned RL framework. This chapter summarizes the conducted experiments and outlines the obtained results. Section 7.1 focuses on answering SQ1&2, Section 7.2 discusses experiments and findings regarding SQ3&4, and Section 7.3 is dedicated to answering SQ5.

7.1 Parameter Selection (SQ1&2)

The first set of simulations has been conducted with the goal of answering SQ1, by suggesting a number of extinction trials n available for the RL agent to robustly and efficiently extinguish a patient's fear, and SQ2, by finding a prior for the RW model learning rate α that captures empirically reasonable speed of human fear learning and that supports the RL agent to find optimized ET protocols. SQ1 and SQ2 have been studied together, as the choice on one of these SQs' parameter influences the other's parameter, which will become clearer in the course of this section.

In the experiments, the numbers of extinction trials $n \in [24, 32, 64]$ and the priors $\alpha \in [\mathcal{U}(0, 0.2), \mathcal{U}(0, 0.5), \mathcal{U}(0, 1)]$ have been considered. These choices origin from preceding considerations that are discussed in the following.

Preceding Considerations

The starting point for the number of stage 2 extinction trials was 64, to coincide with the number of acquisition trials in stage 1. Simulations using 64 extinction trials have shown that – depending on the patient's learning rate α – not all of the trials might be required for extinguishing the acquired fear. On the other hand, if α is considerably high, already a few extinction trials suffice for a total fear extinction. In such cases, the RL agent can choose inefficient ET protocols such as e.g., first strengthening the fear by negatively reinforcing CS+ and afterwards fully extinguishing it. The described protocol has no negative effect on the reward, as only fear levels in stage 3 (RET) are of relevance. Still, this is not an optimized ET protocol due to reasons of inefficiency, caused by a surplus of extinction trials the RL agent can "play" around with. These findings encouraged the assumption that less extinction trials might lead to more efficient ET protocols. Too few extinction trials on the other hand might not suffice to fully extinguish the fear. With the goal of finding a compromise between the two scenarios, the mentioned numbers of extinction trials $n \in [24, 32, 64]$ have been tested.

To answer SQ2, different priors have been considered for the RW model learning rate α, which is a crucial parameter for the magnitude of weight updates in the RW model. Being restricted to $\alpha \in [0, 1]$, the first seemingly obvious approach was to start with the standard uniform distribution $\alpha \sim \mathcal{U}(0, 1)$. Test simulations have shown that for learning rates close to 1, a small number of trials (e.g., <3) might suffice to fully extinguish a fear. Depending on the number of extinction trials, this can lead to redundant trials potentially causing inefficient ET protocols, as mentioned in the last paragraph. Also, α values close to 1 along with their rapid fear extinction do not seem to reflect the empirically complex phenomenon of fear extinction [96]. In order to find a prior for α that reflects reasonable human speed of learning, while supporting the optimization of ET protocols, besides $\alpha \sim \mathcal{U}(0, 1)$, two further – more restricted – priors have been considered in following simulations: $\alpha \sim \mathcal{U}(0, 0.2)$ and $\alpha \sim \mathcal{U}(0, 0.5)$. The two ranges have been chosen heuristically to represent one moderately restricted and one rather tightly restricted prior for the learning rate α. Another aim of trying different α priors is to find a combination of extinction trials n and α prior that enable the RL agent to find optimized ET protocols.

Experiments

20 simulations have been conducted with each combination of the different numbers of extinction trials $n \in [24, 32, 64]$ and priors for $\alpha \in [\mathcal{U}(0, 0.2), \mathcal{U}(0, 0.5), \mathcal{U}(0, 1)]$. In these simulations, reward function absSummedRETbasedReward (see Equation (5.29)) has been used with the linear observation function (see Equation (5.21)). Each model has been trained for 100000 steps with evaluation callbacks at each 1000 training steps for

Table 7.1: Setup for the simulations discussed in the present paragraph.

Setup	
Patient model	RW
Reward function	absSummedRETbasedReward
Observation function	linear
RecurrentPPO hyperparameters	default

Table 7.2: Different combinations for the number of extinction trials n and the prior for the learning rate α with resulting [1,0,0]- and all-[1,0,0]-fractions. The combinations with [1,0,0]-fractions>0.8 are highlighted in blue.

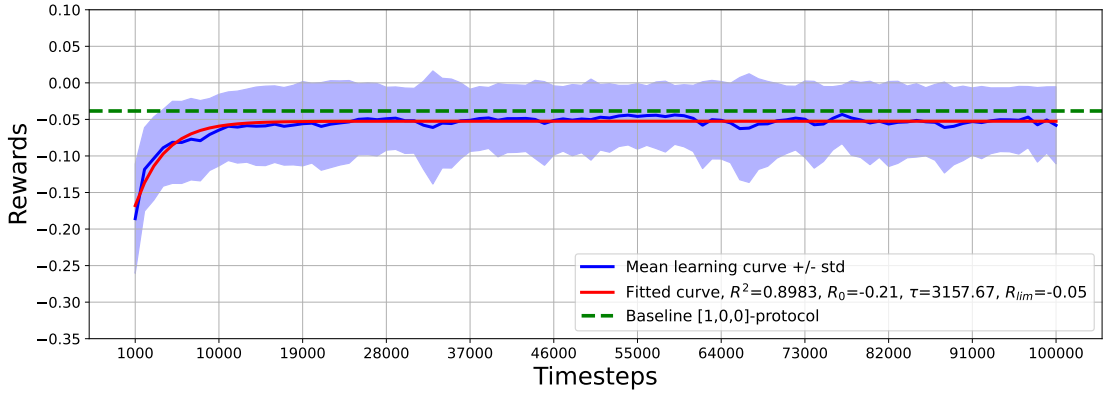
Extinction Trials	**Prior for α**	**[1,0,0]-fraction**	**all-[1,0,0]-fraction**
24	$\alpha \sim \mathcal{U}(0, 0.2)$	0.8970	0.7390
	$\alpha \sim \mathcal{U}(0, 0.5)$	0.8604	0.6985
	$\alpha \sim \mathcal{U}(0, 1)$	0.6499	0.6186
32	$\alpha \sim \mathcal{U}(0, 0.2)$	0.9088	0.8421
	$\alpha \sim \mathcal{U}(0, 0.5)$	0.7505	0.6657
	$\alpha \sim \mathcal{U}(0, 1)$	0.5166	0.5151
64	$\alpha \sim \mathcal{U}(0, 0.2)$	0.6007	0.2725
	$\alpha \sim \mathcal{U}(0, 0.5)$	0.5477	0.5000
	$\alpha \sim \mathcal{U}(0, 1)$	0.3523	0.2735

100 episodes using default hyperparameters for the RecurrentPPO model. The simulation setup for these initial experiments is summarized in Table 7.1.

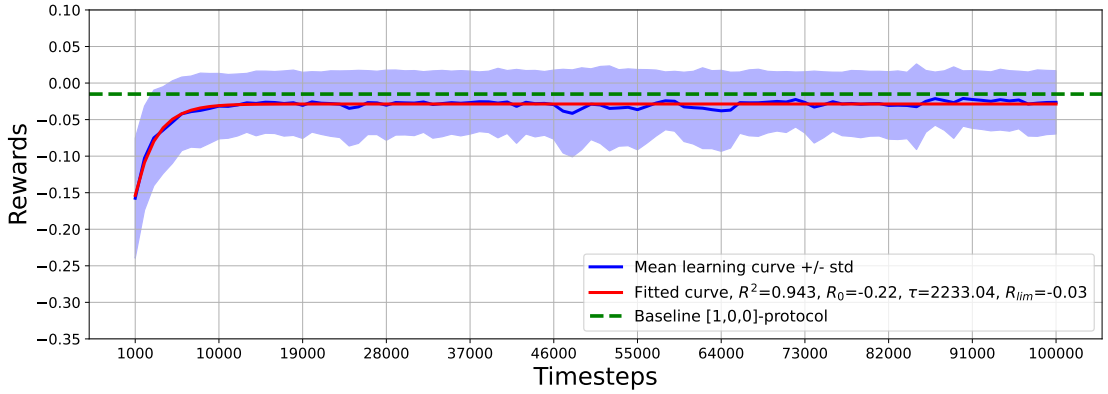
Based on Equation (5.36), a mean learning curve has been calculated from the 20 simulations per combination of extinction trials and α priors, and a curve has been fitted as described in Section 5.5. The resulting learning curves are shown in Figures 7.1-7.3, with the legend describing R^2 and the curve fitting parameters (for more detailed description see Section 5.5).

When using the RW patient model, the ET protocol leading to the fastest fear extinction is the [1,0,0]-protocol, i.e. only CS+-alone presentations. To check which of the 20 trained RL models per combination of extinction trials and α priors learned the aimed-for ET protocol, the [1,0,0]-fraction (see Equation (5.33)) and the all-[1,0,0]-fraction (see Equation (5.34)) have been calculated based on 1000 evaluation episodes per trained model with deterministic actions.

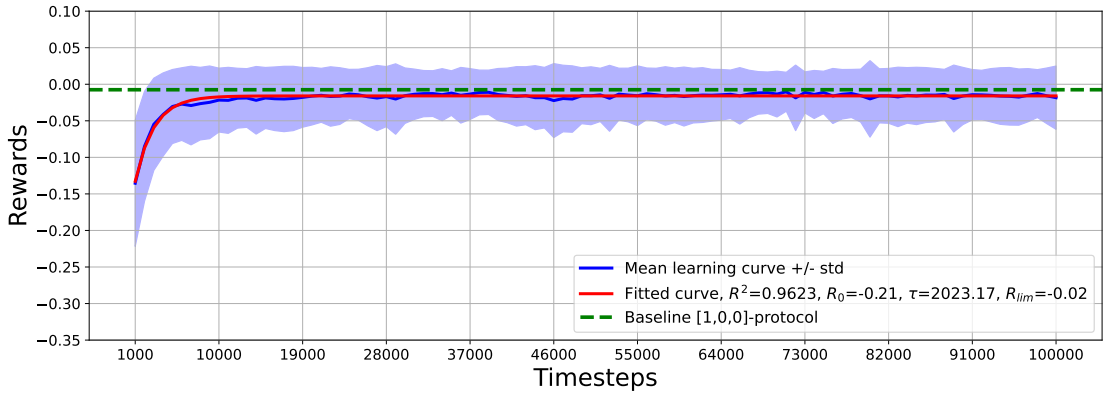
Table 7.2 lists the obtained [1,0,0]- and all-[1,0,0]-fractions for the different combinations (see also Figure 7.4). At this stage of the experiments the focus was on finding an ET protocol that mainly contains CS+-only presentations rather then finding the all-[1,0,0]-actions protocol. Therefore, the combinations with [1,0,0]-fraction of > 0.8 (highlighted in blue) will be subject of further investigations.



(a) 24 extinction trials with $\alpha \sim \mathcal{U}(0, 0.2)$.

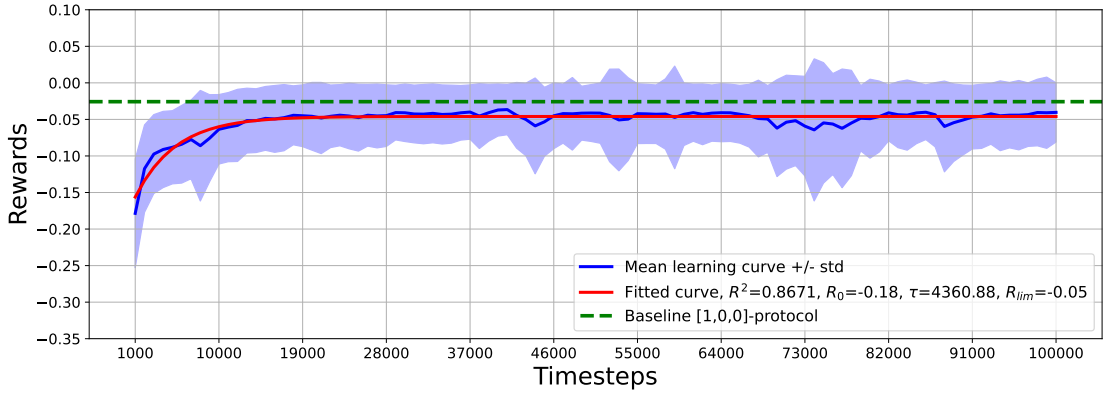


(b) 24 extinction trials with $\alpha \sim \mathcal{U}(0, 0.5)$.

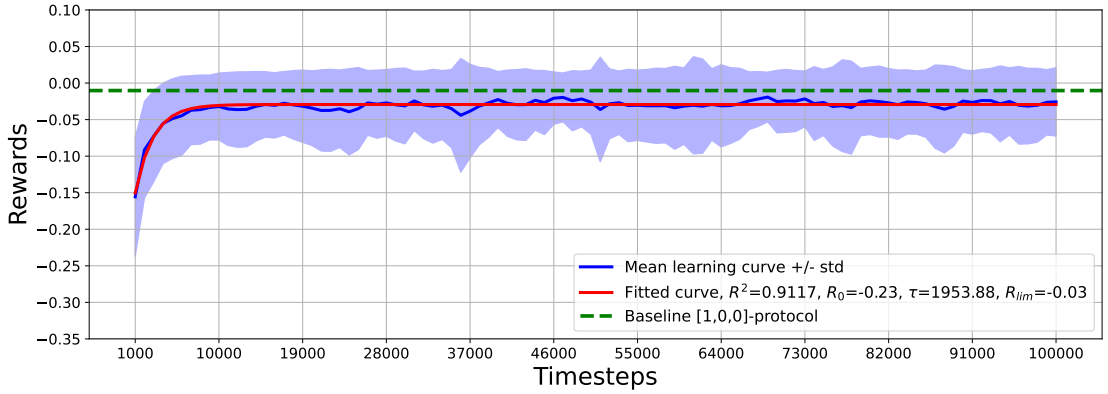


(c) 24 extinction trials with $\alpha \sim \mathcal{U}(0, 1)$.

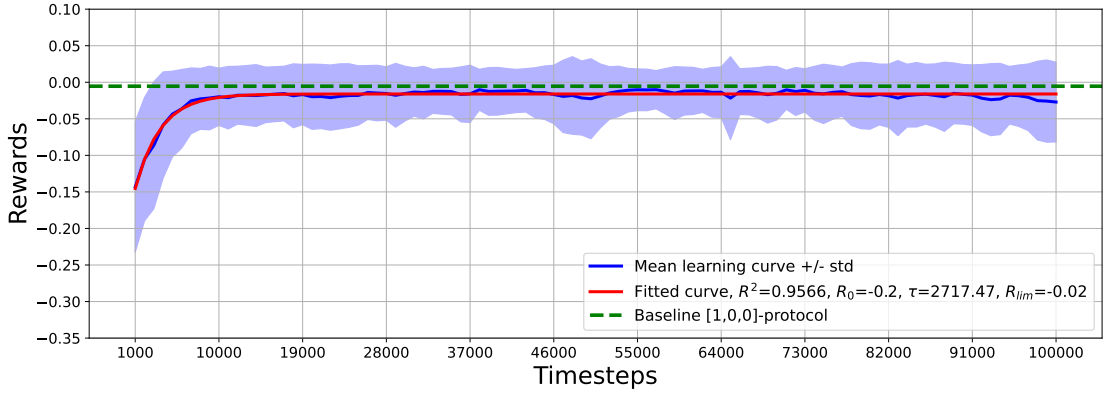
Figure 7.1: Mean and fitted learning curve for 24 extinction trials using (a) $\alpha \sim \mathcal{U}(0, 0.2)$, (b) $\alpha \sim \mathcal{U}(0, 0.5)$, and (c) $\alpha \sim \mathcal{U}(0, 1)$ based on 20 simulations with evaluation callbacks each 1000 training steps for 100 episodes. Shown are the mean rewards of the evaluation callbacks averaged over the 20 simulations (bold blue line), the standard deviation (light blue area), the fitted curve (red line) and a baseline showing the average reward over 10000 patients when applying the [1,0,0]-protocol (green dashed line). All three learning curves show an increase over the training period and almost approach the [1,0,0]-baseline.



(a) 32 extinction trials with $\alpha \sim \mathcal{U}(0, 0.2)$.

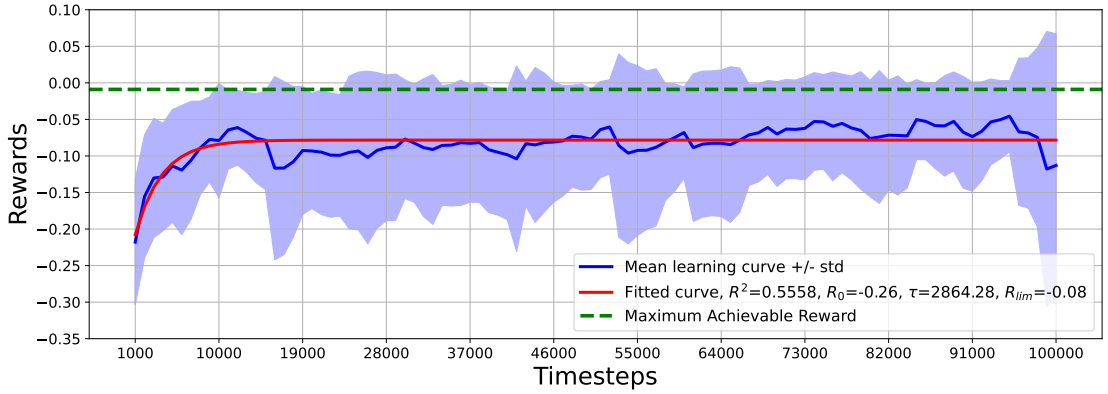


(b) 32 extinction trials with $\alpha \sim \mathcal{U}(0, 0.5)$.

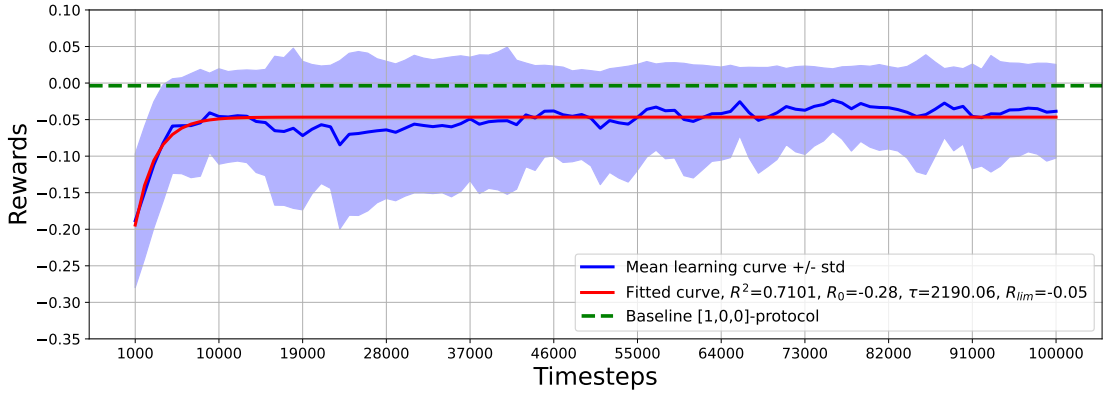


(c) 32 extinction trials with $\alpha \sim \mathcal{U}(0, 1)$.

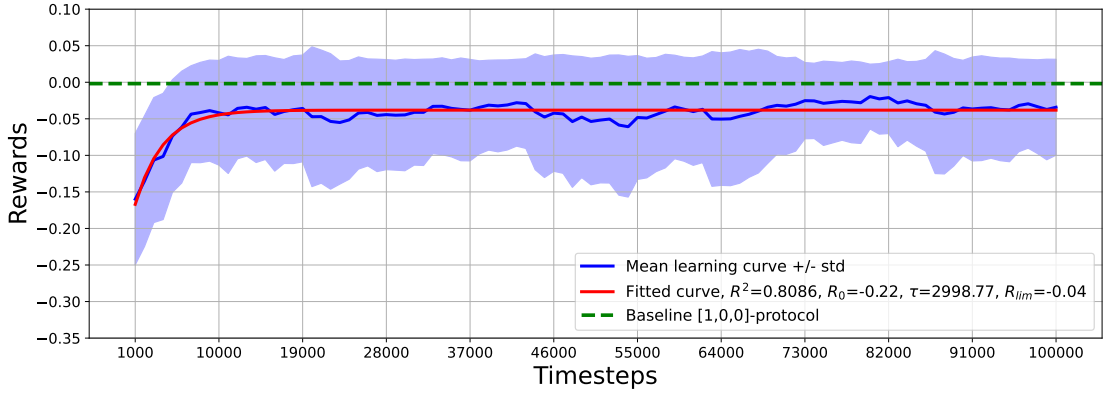
Figure 7.2: Mean and fitted learning curve for 32 extinction trials using (a) $\alpha \sim \mathcal{U}(0, 0.2)$, (b) $\alpha \sim \mathcal{U}(0, 0.5)$, and (c) $\alpha \sim \mathcal{U}(0, 1)$ based on 20 simulations with evaluation callbacks each 1000 training steps for 100 episodes. Shown are the mean rewards of the evaluation callbacks averaged over the 20 simulations (bold blue line), the standard deviation (light blue area), the fitted curve (red line) and a baseline showing the average reward over 10000 patients when applying the [1,0,0]-protocol (green dashed line). All three learning curves show an increase over the training period and almost approach the [1,0,0]-baseline.



(a) 64 extinction trials with $\alpha \sim \mathcal{U}(0, 0.2)$.



(b) 64 extinction trials with $\alpha \sim \mathcal{U}(0, 0.5)$.



(c) 64 extinction trials with $\alpha \sim \mathcal{U}(0, 1)$.

Figure 7.3: Mean and fitted learning curve for 64 extinction trials using (a) $\alpha \sim \mathcal{U}(0, 0.2)$, (b) $\alpha \sim \mathcal{U}(0, 0.5)$, and (c) $\alpha \sim \mathcal{U}(0, 1)$ based on 20 simulations with evaluation callbacks each 1000 training steps for 100 episodes. Shown are the mean rewards of the evaluation callbacks averaged over the 20 simulations (bold blue line), the standard deviation (light blue area), the fitted curve (red line) and a baseline showing the average reward over 10000 patients when applying the [1,0,0]-protocol (green dashed line). All three learning curves show an increase over the training period while they plateau at a lower level compared to the [1,0,0]-baseline.

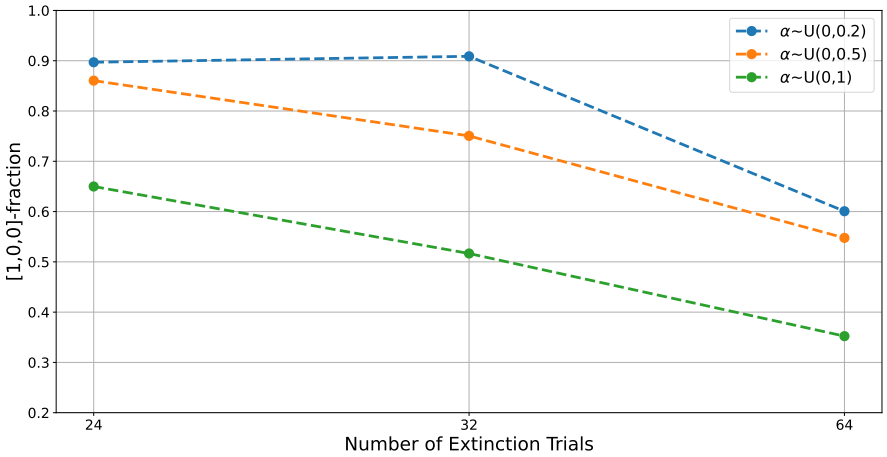


Figure 7.4: [1,0,0]-fractions of the different combinations of extinction trials and α priors. The best results were achieved with 3 combinations: 24 extinction trials and a prior of $\alpha \sim \mathcal{U}(0, 0.2)$, 24 extinction trials and a prior of $\alpha \sim \mathcal{U}(0, 0.5)$ and 32 extinction extinction trials and a prior of $\alpha \sim \mathcal{U}(0, 0.2)$.

An average of >80% [1,0,0]-actions were reached with 3 combinations: 24 extinction trials and a prior of $\alpha \sim \mathcal{U}(0, 0.2)$, 24 extinction trials and a prior of $\alpha \sim \mathcal{U}(0, 0.5)$ and 32 extinction extinction trials and a prior of $\alpha \sim \mathcal{U}(0, 0.2)$.

It might seem counter intuitive that the highest all-[1,0,0]-fraction was achieved with 32 extinction trials, as it seems less likely to have all actions being [1,0,0] for 32 extinction trials than for 24 extinctions trials. But the higher number of extinction trials might have helped the RL agent to learn an ET protocol with high action probabilities for the [1,0,0]-action and as deterministic actions have been used in the evaluations, i.e. choosing the action with the highest probability, this might have led to many all-[1,0,0]-protocols and therefore high all-[1,0,0]-fractions.

To decide between 24 and 32 extinction trials, the two options have been further examined by learning curve analyses. The learning curves have been analyzed as described in Section 5.5, where an exponential-growth-to-a-limit curve has been fitted to the mean learning curves. Table 7.3 shows the fitting parameters R_{lim} and τ as well as the number of patients required during training for reaching the asymptotic performance R_{lim}. As described in Section 5.5, 5τ gives the number of training steps required for reaching $\sim 99.3\%$ of asymptotic performance. The number of patients required for reaching this performance is calculated by $\frac{5\tau}{\text{extinction trials}}$ (see Equation (5.39)). Table 7.3 lists the results from the curve fitting process. The best result is highlighted in blue and was achieved with 24 extinction trials and a prior of $\alpha \sim \mathcal{U}(0, 0.5)$.

Decision on Parameters

Although both, 24 and 32 extinction trials, would be reasonable choices to proceed with in ongoing simulations, the smallest τ value was achieved using 24 trials which encouraged

Table 7.3: Results of mean learning curve analysis. Only the settings with the most promising preceding results were considered. Highlighted in blue is the smallest number of patients required for reaching $\sim 99.3\%$ of asymptotic performance.

Extinction Trials	**Prior for α**	R_{lim}	τ	**Required Patients** $(\frac{5\tau}{\text{extinction trials}})$
24	$\alpha \sim \mathcal{U}(0, 0.2)$	-0.05	3157.67	658
	$\alpha \sim \mathcal{U}(0, 0.5)$	-0.03	2233.04	466
32	$\alpha \sim \mathcal{U}(0, 0.2)$	-0.05	4360.88	682

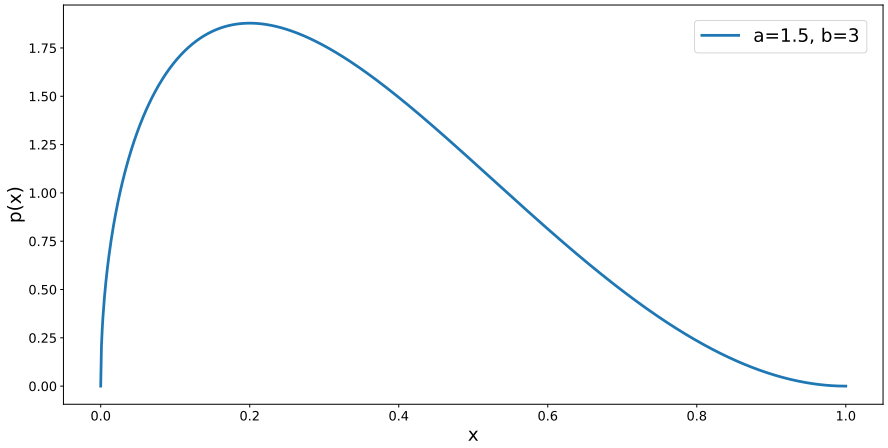


Figure 7.5: Prior for the RW model's learning rate $\alpha \sim \mathcal{B}(1.5, 3)$, with mean $\mu = 0.3333$, standard deviation $\sigma = 0.2010$ and variance $\sigma^2 = 0.0404$.

the decision to continue with 24 extinction trials in the proceeding simulations.

Regarding the RW model learning rate α, the highest [1,0,0]-fraction was achieved using $\mathcal{U}(0, 0.2)$ and $\mathcal{U}(0, 0.5)$. The α prior used in the proceeding simulations should therefore reflect that smaller α ranges have led to higher [1,0,0]-fractions, while also considering that $\alpha \in [0, 1]$. One example for a distribution capturing both properties is $\alpha \sim \mathcal{B}(1.5, 3)$ (see Figure 7.5), which is in line with learning rate estimates of empirical human learning experiments [128, 15] (see Section 8.1 for further discussion). Hence, in ongoing simulations, $\alpha \sim \mathcal{B}(1.5, 3)$ is the choice for the RW learning rate α prior. With a mean of $\mu = 0.3333$, smaller values for α are sampled with higher probability, while still considering the possibility of high α values up to $\alpha = 1$.

7.2 Influence of Reward Funtions and Observation Functions (SQ3&4)

Next, the focus was on answering SQ3, by using different reward functions to calculate the feedback for the RL agent and SQ4, by using different patient model observation functions in the simulations.

For each simulation setup, i.e. a specified patient model, reward function, observation function and hyperparameters for the RecurrentPPO model, 20 simulations have been conducted for 100000 training steps each. In order to assess whether a setup has led to a successful learning, the 2-step evaluation procedure described in Section 5.5 was applied to the different models. Each trained model was evaluated with 1000 evaluation episodes and the following fractions were calculated: 1. The [1,0,0]-fraction (see Equation (5.33)) describing how many of the 100 episodes $\cdot$ 24 actions = 24000 actions were the CS+-only action and 2. the all-[1,0,0]-fraction (see Equation (5.34)) describing in how many of the 1000 episodes all 24 actions were the CS+-only action. As for answering SQ3 and SQ4 all simulations have been conducted with the RW patient model, the [1,0,0]-protocol is the aimed-for outcome in all of Section 7.2's simulations. Additionally, in the promising setups (high all-[1,0,0]- and/or [1,0,0]-fractions) an exponential growth-to-the-limit curve was fitted to the learning curves (see Section 5.5).

In the following, different combinations of patient model observations functions, reward functions and RecurrentPPO hyperparameters have been analyzed and compared amongst each other with the goal of finding experimental setups that lead to robust and efficient ET protocols. The individual subsections contain a table outlining the simulation setup (patient model, observation function, reward function, RecurrentPPO hyperparameters), a mean learning curve, the achieved results ([1,0,0]-fraction, all-[1,0,0]-fraction and optionally curve-fitting parameters) and the common ET protocols suggested by the RL agent using the specified simulation setup.

Linear Observation Function, Mean of CS+'s CRs as Reward

As a starting point, reward function RETbasedReward (see Equation (5.27)), which calculates the mean of CRs to CS+ presented during the RoF-test, has been used in the simulations. The simulations have been conducted with the linear observation function, the RW model as stand-in for the patient, and default hyperparameters for the RecurrentPPO model (see Table 7.4 for setup overview).

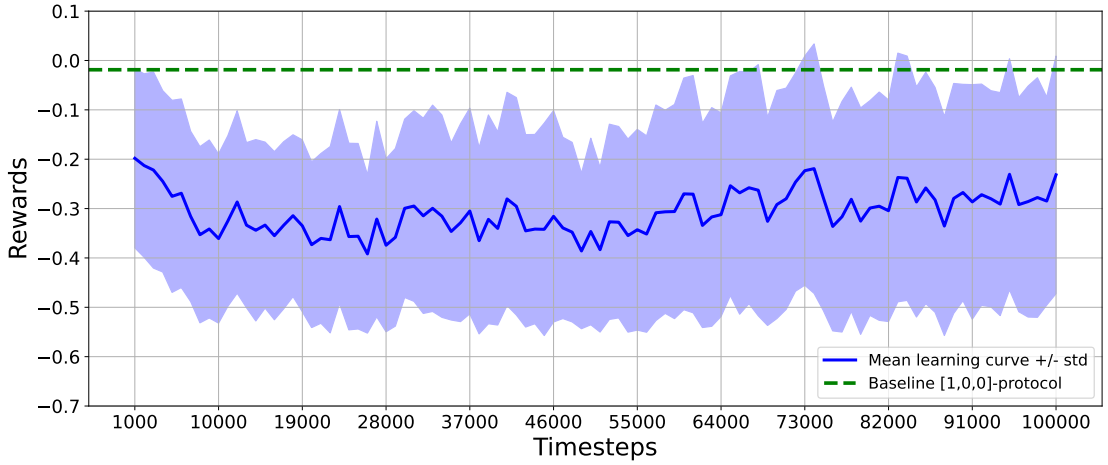


Figure 7.6: Mean learning curve of 20 simulations with the RW model, linear observation function, reward function RETbasedReward (Equation (5.27)), and default hyperparameters. Shown are the mean rewards of the evaluation callbacks each 1000 training steps for 100 episodes averaged over the 20 simulations (bold blue line), the standard deviation (light blue area), and a baseline showing the average reward over 10000 patients when applying the [1,0,0]-protocol (green dashed line). The learning curve does not significantly increase during the training process and stays at a lower level compared to the [1,0,0]-baseline.

Table 7.4: Setup and results for the simulations discussed in the present subsection.

Setup	
Patient model	RW
Observation function	linear
Reward function	RETbasedReward
RecurrentPPO hyperparameters	default
Results	
[1,0,0]-fraction	0.0313
all-[1,0,0]-fraction	0

Figure 7.6 shows the mean learning curve with standard deviation averaged over the 20 simulations for this paragraph's simulation setup and a baseline showing the average reward over 10000 patients when applying the [1,0,0]-protocol. The learning curve does not significantly increase during the training process and stays at a lower level compared to the [1,0,0]-baseline. With a [1,0,0]-fraction of 0.0313 and an all-[1,0,0]-fraction of 0, this simulation setup did not lead to ET protocols consisting of mainly CS+-only presentations.

Figure 7.7 shows the three ET protocols regularly chosen by the RL agent using the

simulation setup specified in Table 7.4. Figure 7.7a shows the first of the chosen ET protocols with all actions being [0,1,1], which leads to an increase of CS-'s weights with no effect on the weights of CS+.

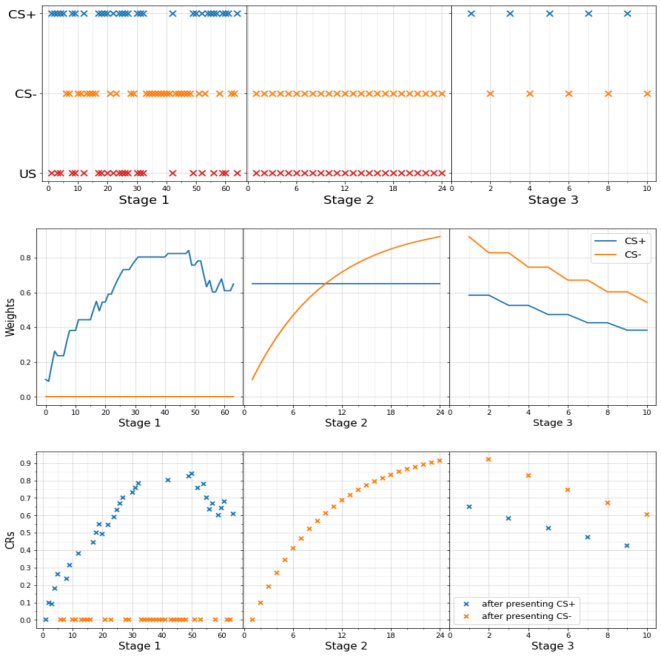
The second frequently chosen ET protocol is shown in Figure 7.7b with all actions being [1,1,0]. This protocol leads to a decrease of both CSs' weights, due to a phenomenon called *conditioned inhibition* [103, 113]. In conditioned inhibitions trials of reinforcing CS+ are interspersed with presentations of the stimulus compound CS+ and CS- without reinforcement, resulting in a negative associative strength attributed to CS-. This negative association is acquired due to the negative prediction error on the compound trials, leading to the assumption that CS- must have a negative weight in order to counterbalance the excitatory weight of CS+. This ET protocol has the effect of decreasing CS+'s weights, but the agent does not recognize that fear is only present for CS+ and that the most efficient strategy to extinguish this fear is the [1,0,0]-protocol.

Figure 7.7c shows the third frequently chosen ET protocol. In this protocol, after some [1,0,0]-actions, a fear for CS- is acquired by presenting CS- with subsequent negative reinforcement. Afterwards, CS+ is shown together with CS- without negative reinforcement ([1,1,0]-action), leading to a higher expectation for the US due to the positive weights for both CSs. The prediction error (see Equation (5.13)) is larger, due to the expected US not being presented and therefore the higher weight updates (see Equation (5.2)) lead to a fast decrease of CS+'s and CS-'s weights, even enabling negative weights for CS- due to properties of the linear observation function. Figure 7.7c's approach is rather refined and exploits some characteristics of the RW model, however it significantly differs from the aimed-for [1,0,0]-protocol.

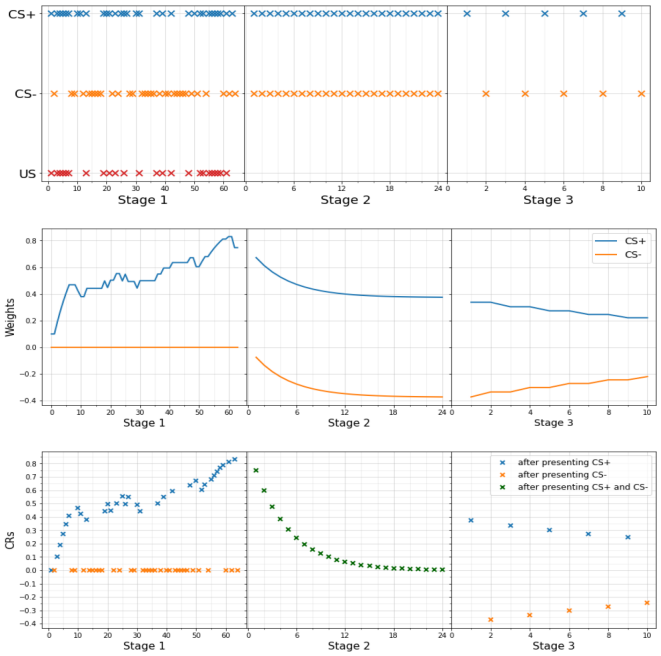
While the ET protocols in Figure 7.7b-c lead to a desired decrease of CS+'s weights and thus increasing rewards, this is not the case for Figure 7.7a, where it is not clear why the RL agent did not recognize that it is favorable to lower CS+'s weights.

Linear Observation Function, Mean of Both CSs' CRs as Reward

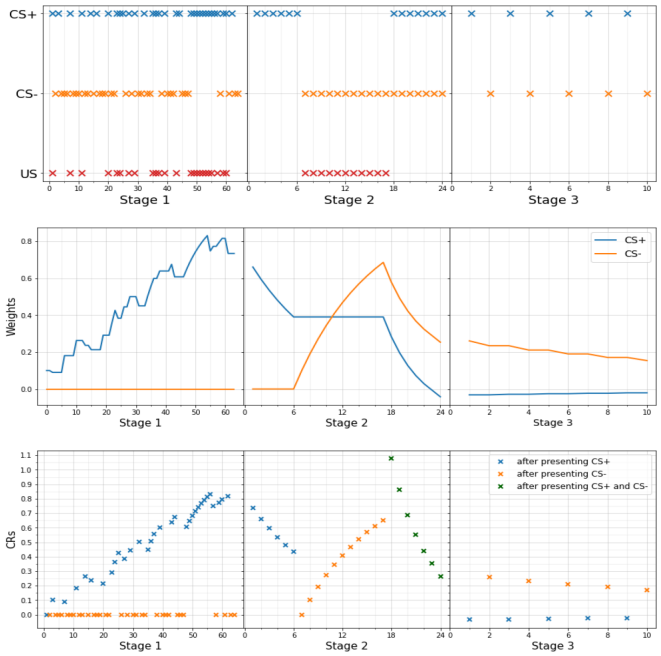
Next, reward function SummedRETbasedReward, which calculates the mean of the CRs to both CSs presented during RoF-testing (see Equation (5.27)) has been used as reward for the RL agent with the RW model as stand-in for the patient. The simulations have been conducted using the linear observation function and default hyperparameters for the RecurrentPPO model (see Table 7.5 for setup overview).



(a) ET protocol suggested by several of the 20 trained RL models, consisting of [0,1,1]-actions only, which leads to an increase of CS-'s weights while keeping CS+'s weights constant.



(b) ET protocol suggested by several of the 20 trained RL models, consisting of [1,1,0]-actions only, which leads to a decrease of both CSs' weights, resulting in negative weights and CRs for CS- due to conditioned inhibition.



(c) ET protocol suggested by several of the 20 trained RL models, consisting of [1,0,0]-, [0,1,1]-, and [1,1,0]-actions, where during the final [1,1,0]-actions the weights for both CSs drop due to an over-expectation caused by the previous [0,1,1]-actions.

Figure 7.7: ET protocols (a)-(c) suggested by RL models trained with the RW model, linear observation function, reward function RETBASEDREWARD (see Equation (5.27)), and default hyperparameters. For all protocols, time courses of presented stimuli, evolution of weights and evolution of CRs are shown using an RW learning rate of $\alpha = 0.1$.

Table 7.5: Setup and results for the simulations discussed in the present subsection.

Setup	
Patient model	RW
Observation function	linear
Reward function	SummedRETbasedReward
RecurrentPPO hyperparameters	default
Results	
[1,0,0]-fraction	0.6621
all-[1,0,0]-fraction	0

Figure 7.8 shows the mean learning curve with standard deviation averaged over the 20 simulations for this paragraph's simulation setup and a baseline indicating the average reward over 10000 patients when applying the [1,0,0]-protocol. The mean learning curve increases over the training period and eventually exceeds the [1,0,0]-baseline, while the higher rewards do not translated into lowered fear, but are caused by enforcing negative weights for CS- (see ET protocols in Figure 7.9).

With a [1,0,0]-fraction of 0.6621 and an all-[1,0,0]-fraction of 0, this simulation setup has more often led to the choice of [1,0,0]-actions than last section's setup, but the ET protocol with all [1,0,0]-actions has not been found.

Figure 7.9 shows the two ET protocols that have been regularly chosen by the RL agent using the simulation setup specified in Table 7.5. The ET protocol shown in Figure 7.9a first uses some [1,1,0]-actions and for all remaining trials the [1,0,0]-action. Initially, this leads to a decrease of both CSs' weights, causing the weights of CS- to become negative due to properties of the linear observation function. Later on, with the [1,0,0]-actions, only the weights of CS+ continue to decrease. The agent's reason for initially choosing [1,1,0] instead of [1,0,0] lies in the reward function, which calculates the mean of both CSs' weights leading to better results if some of the weights are negative.

The second frequently chosen ET protocol (shown in Figure 7.9b) aims at the same strategy (conditioned inhibition) with all actions being [1,1,0]. This ET protocol has the effect of decreasing both CSs' weights, and therefore leading to high rewards using reward function SummedRETbasedReward (see Equation (5.28)).

While both frequently found ET protocols lead to a decrease of the patient's fear, they differ from the most efficient protocol for the RW patient model, which only focuses on lowering the expected US of the CS with positive weights. When using the mean of both CSs' weights as reward function (SummedRETbasedReward), the agent exploits the possibility to reach negative weights with the linear observation function in order to achieve higher rewards. To prevent the agent from this behavior, in a following step a reward function has been used that considers absolute values of the CSs' weights.

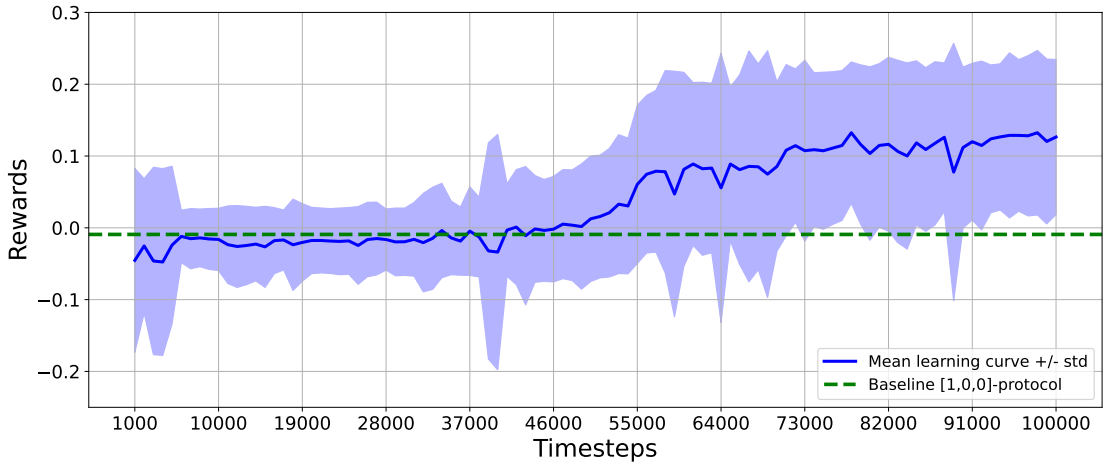
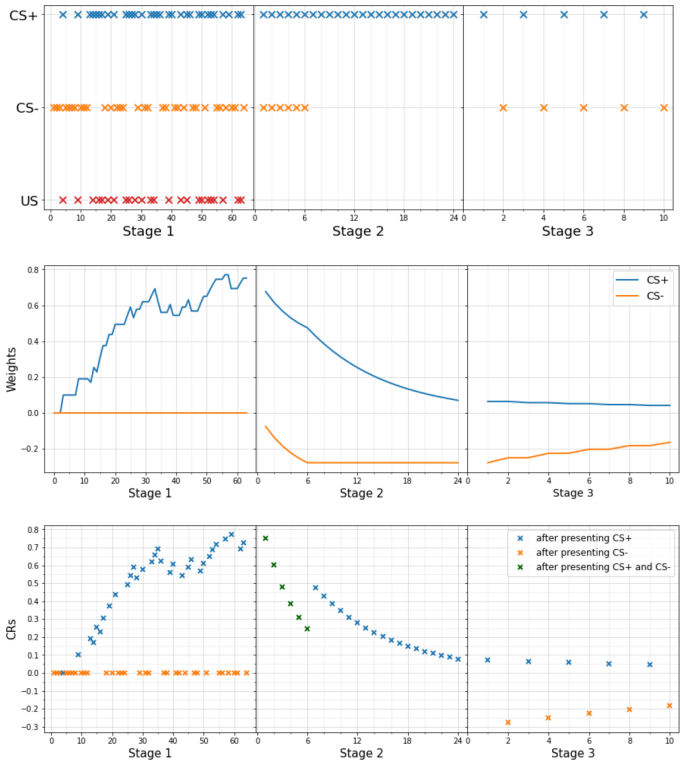
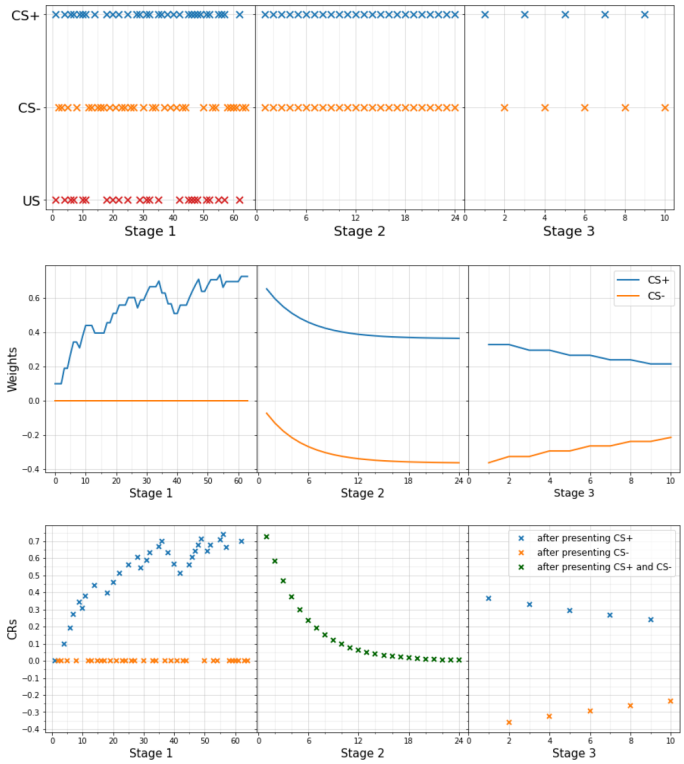


Figure 7.8: Mean learning curve of 20 simulations with the RW model, linear observation function, reward function SummedRETbasedReward (see Equation (5.28)), and default hyperparameters. Shown are the mean rewards of the evaluation callbacks each 1000 training steps for 100 episodes averaged over the 20 simulations (bold blue line), the standard deviation (light blue area), and a baseline showing the average reward over 10000 patients when applying the [1,0,0]-protocol (green dashed line). The mean learning curve increases over the training period and eventually exceeds the [1,0,0]-baseline, while the higher rewards do not translate into lowered fear, but are caused by enforcing negative weights for CS-.



(a) ET protocol suggested by several of the 20 trained RL models, consisting of [1,1,0]-actions, which lead to decreasing weights for both CSs due to conditioned inhibition, and [1,0,0]-actions, which further decrease the weights of CS+.



(b) ET protocol suggested by several of the 20 trained RL models, consisting of [1,1,0]-actions only, which lead to decreasing weights for both CSs due to conditioned inhibition.

Figure 7.9: ET protocols (a)-(b) suggested by RL models trained with the RW model, linear observation function, reward function SummedRETBasedReward (see Equation (5.28)), and default hyperparameters. For all protocols, time courses of presented stimuli, evolution of weights and evolution of CRs are shown using an RW learning rate of $\alpha = 0.1$.

Linear Observation Function, Mean of Both CSs' Absolute CRs as Reward

In order to prevent the agent from lowering CS-'s weights leading to negative CRs, reward function absSummedRETbasedReward (see Equation (5.29)) has been used, where the CRs' absolute values are considered. The simulations have been conducted using the RW model, linear observation function, and default hyperparameters for the RecurrentPPO model (see Table 7.6 for setup overview).

Table 7.6: Setup and results for the simulations discussed in the present subsection.

Setup	
Patient model	RW
Observation function	linear
Reward function	absSummedRETbasedReward
RecurrentPPO hyperparameters	default
Results	
[1,0,0]-fraction	0.8150
all-[1,0,0]-fraction	0.7371
fitting parameters	R_{lim}= -0.04 τ= 10119.6 R^2 = 0.7585

Figure 7.10 shows the mean and fitted learning curve with standard deviation averaged over the 20 simulations for this paragraph's simulation setup and a baseline showing the average reward over 10000 patients when applying the [1,0,0]-protocol. The mean learning curve is increasing to a level slightly lower than the [1,0,0]-baseline, where it remains for the rest of the training process. Using reward function absSummedRETbasedReward, not only the [1,0,0]-fraction with 0.8150 is the highest up to this point, but also the all-[1,0,0]-fraction with 0.7371 is relatively high and the first one differing from 0 in the conducted simulations.

Figure 7.9 shows the two ET protocols regularly chosen by the RL agent using the simulation setup specified in Table 7.6. One is the aimed-for [1,0,0]-protocol with all actions being CS+-only presentations (see Figure 7.11a). This is the optimal scenario, where the agent recognizes that fear is only related to CS+, and it chooses the most efficient protocol for breaking the association between CS+ and the negative reinforcement. When looking at the evolution of weights during stage 2 in Figure 7.11a it can be seen, that the weights for CS+, which have increased during stage 1, decrease to a level close to 0, while CS-'s weights stay 0 throughout all stages.

As expressed in the [1,0,0]-fraction and the all-[1,0,0]-fraction, not all of the 20 trained RL models found the aimed-for ET protocol with this section's simulation setup. In 4 of the 20 simulations, the RL agent learns a protocol with all actions being [0,1,0] (see Figure 7.11b). This ET protocol does neither effect the weights for CS+ nor the weights for CS- during stage 2 and therefore fails to extinguish the fear associated with CS+.

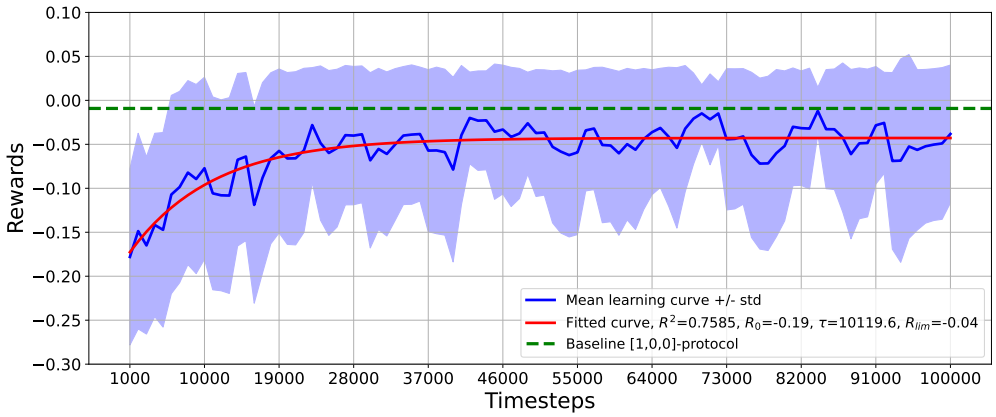
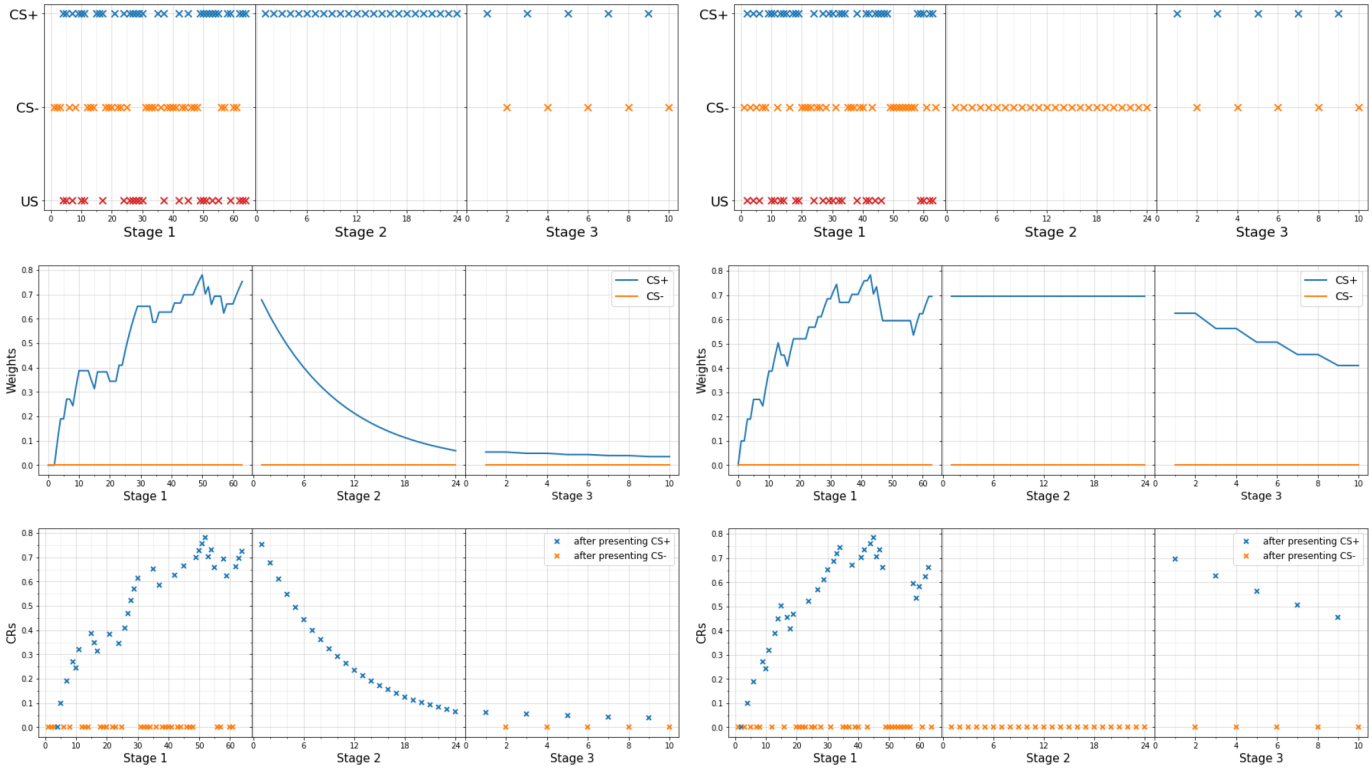


Figure 7.10: Mean and fitted learning curve of 20 simulations with the RW model, linear observation function, reward function absSummedRETbasedReward (see Equation (5.29)), and default hyperparameters. Shown are the mean rewards of the evaluation callbacks each 1000 training steps for 100 episodes averaged over the 20 simulations (bold blue line), the standard deviation (light blue area), the fitted curve (red line), and a baseline showing the average reward over 10000 patients when applying the [1,0,0]-protocol (green dashed line). The mean learning curve is increasing to a level slightly lower than the [1,0,0]-baseline, where it remains for the rest of the training process.

Due to the well-behaving learning curve (see Figure 7.10) and the high values for the [1,0,0]- and all-[1,0,0]-fraction, an exponential curve has been fitted to the mean learning curve of the 20 simulations based on Section 5.5. The resulting parameters from the fitting process are R_{lim}= 0.04, τ = 10119.6, and R^2 = 0.7585 (see also Table 7.6). Based on Equation (5.39) the number of patients (corresponding to RL episodes) required for reaching asymptotic performance is $\frac{5\tau}{\text{extinction trials}} = \frac{5*10119.6}{24} \sim 2109$ patients.

Sigmoid Observation Function, Non-differential Reward Function with ACQ Correction

The above subsection has shown that using absolute CRs makes lowering CS-'s weights no longer profitable. Another way to achieve the same effect is to use a sigmoid observation function as opposed to the linear one. This puts a limit on the benefit which comes with lowering CS-'s weights. The following simulations use the RW model as stand-in for the patient, the sigmoid observation function, and default hyperparameters for the RecurrentPPO model (see Table 7.7 for setup overview). The simulations have been conducted using reward function ACQcorrectedNonDifferentialReward (see Equation (5.30)), with which not only CS+'s CRs during the RoF-testing (stage 3) are taken into account, but also CS+'s CRs at the end of ACQ (stage 1). The resulting reward expresses the percentage of fear extinguished after stage 2 compared to the fear present at the end of stage 1.



(a) ET protocol suggested by several of the 20 trained RL models, consisting of [1,0,0]-actions only, which leads to decreasing weights for CS+ while keeping CS-'s weights constant and which is the aimed-for ET protocol for the RW patient model.

(b) ET protocol suggested by several of the 20 trained RL models, consisting of [0,1,0]-actions only, which does not have an effect on any of the two CSs' weights.

Figure 7.11: ET protocols (a)-(b) suggested by RL models trained with the RW model, linear observation function, reward function absSummedRETbasedReward (see Equation (5.29)), and default hyperparameters. For both ET protocols, time courses of presented stimuli, evolution of weights and evolution of CRs are shown using an RW learning rate of $\alpha = 0.1$.

Table 7.7: Setup and results for the simulations discussed in the present subsection.

Setup	
Patient model	RW
Observation function	sigmoid
Reward function	ACQcorrectedNonDifferentialReward
RecurrentPPO hyperparameters	default
Results	
[1,0,0]-fraction	0.8363
all-[1,0,0]-fraction	0.6223

Figure 7.12 shows the mean learning curve with standard deviation averaged over the 20 simulations for this paragraph's simulation setup and a baseline showing the average reward over 10000 patients when applying the [1,0,0]-protocol. After an initial increasing and stable training phase reaching the level of the [1,0,0]- baseline, the mean learning curve drops underneath the [1,0,0]-baseline after around half of the training time. Due to the decreasing learning curve in the second half of the training process, the exponential-rise-to-a-limit curve has not been fitted to the mean learning curve.

Using this section's simulations setup, a [1,0,0]-fraction of 0.8363 and an all-[1,0,0]-fraction of 0.6223 were achieved.

Figure 7.13 shows the two ET protocols regularly chosen by the RL agent using the simulation setup specified in Table 7.7. Using the sigmoid observation function with initial weights of 0 translates into a certain level of fear for both CSs at the start of the simulations, which can be seen in the positive CR levels at the beginning of stage 1 and the decreasing weights for CS- during stage 1, due to the non-reinforced CS- trials during this stage.

The first frequently chosen ET protocol is the aimed-for [1,0,0]-protocol with all actions being CS+-only presentations (see Figure 7.13a). This is the optimal scenario, where the agent recognizes that fear is only present for CS+, and it chooses the most efficient [1,0,0]-protocol to break the association between CS+ and the negative reinforcement.

A second frequently chosen ET protocol is shown in Figure 7.13b, where the agent chooses several [0,1,0]-actions before continuing with the aimed-for [1,0,0]-action. While the agent recognizes the optimal ET protocol after several steps, the overall protocol slightly differs from the most efficient way to diminish fear for CS+.

Looking at the mean learning curve of this paragraph's simulations (see Figure 7.12), a drift after around half of the simulation time can be recognized as discussed above. With the goal of achieving a stable learning curve while increasing the (all-)[1,0,0]-fraction, the hyperparameters of the RecurrentPPO model have been tuned in a next step, using reward function ACQcorrectedNonDifferentialReward (see Equation (5.30)) and the sigmoid observation function (see Section 5.2.4).

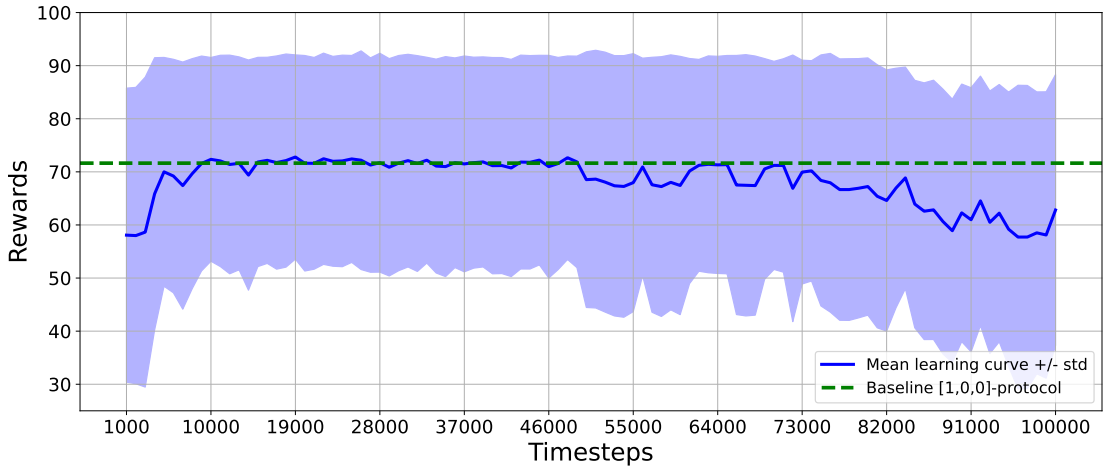
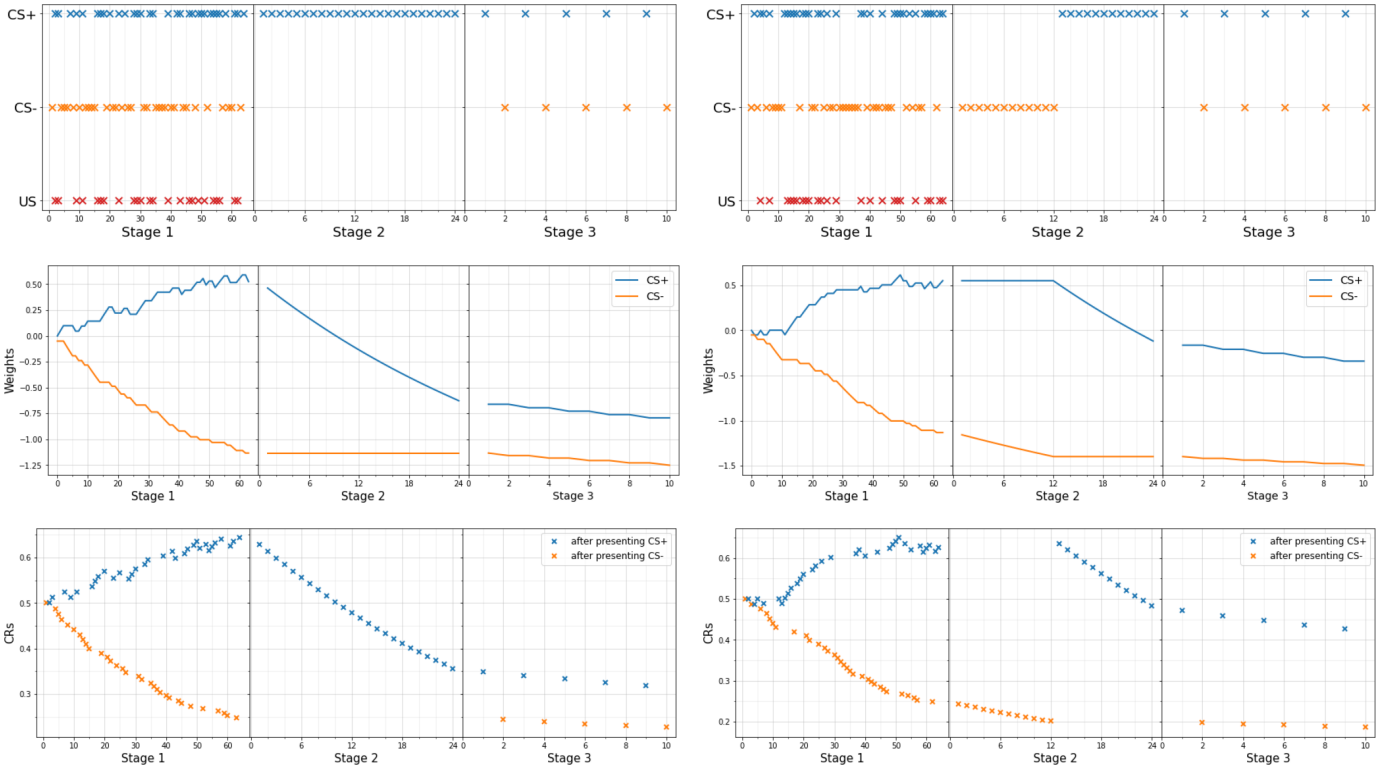


Figure 7.12: Mean learning curve of 20 simulations with the RW model, sigmoid observation function, reward function ACQcorrectedNonDifferentialReward (see Equation (5.30)), and default hyperparameters. Shown are the mean rewards of the evaluation callbacks each 1000 training steps for 100 episodes averaged over the 20 simulations (bold blue line), the standard deviation (light blue area), and a baseline showing the average reward over 10000 patients when applying the [1,0,0]-protocol (green dashed line). After an initial increasing and stable training phase reaching the level of the [1,0,0]-baseline, the mean learning curve drops underneath the [1,0,0]-baseline after around half of the training time.



(a) ET protocol suggested by several of the 20 trained RL models, consisting of [1,0,0]-actions only, which leads to decreasing weights for CS+ while keeping CS-'s weights constant and which is the aimed-for ET protocol for the RW patient model.

(b) ET protocol suggested by several of the 20 trained RL models, consisting of [0,1,0]- and [1,0,0]-actions, first leading to a decrease of CS-'s weights and then to a decrease of CS+'s weights.

Figure 7.13: ET protocols (a)-(b) suggested by RL models trained with the RW model, sigmoid observation function, reward function ACQCORRECTEDNONDIFFERENTIALREWARD (see Equation (5.30)), and default hyperparameters. For both ET protocols, time courses of presented stimuli, evolution of weights and evolution of CRs are shown using an RW learning rate of $\alpha = 0.1$.

Hyperparameter Tuning

So far, different reward and observation functions have been analyzed with some of them leading to promising results, but none of them achieving an all-[1,0,0]-fraction of 1, i.e., enabling the RL agent to find the aimed for ET protocol in all of the simulations. Besides the reward and observations functions, also the hyperparameters of the RecurrentPPO model influence the learning process. To see whether optimized hyperparameters support all-[1,0,0]-fractions close to 1, the hyperparameters of the RL algorithm were tuned and the resulting hyperparameter values used in the proceeding simulations.

For hyperparameter tuning, the RL framework RL Baselines3 Zoo (rl_zoo3) [110] has been utilized, which uses the advanced hyperparameter optimization framework Optuna [3]. Details about the Optuna study, the underlying theory and hyperparameter ranges for the sampling are discussed in Section 5.4 and Section 6.2.

The top two hyperparameter sets found during the Optuna study achieved very similar results with rewards just over 75 using the non-differential reward function with acquisition correction (ACQCORRECTEDNONDIFFERENTIALREWARD, see Equation (5.30)). After some further testing (data not shown), the decision was made for one of the top two hyperparameter sets, which is described below.

Besides selecting optimized hyperparameters, Optuna evaluates the importance of individual hyperparameters for the objective value using the FanovaImportanceEvaluator [65]. For the used simulation setup the *learning_rate* is the dominant hyperparameter having by far the most impact on the objective value (see Figure 7.14), followed by *vf_coef*, *clip_range* and *batch_size*, which are on a similar level amongst each other.

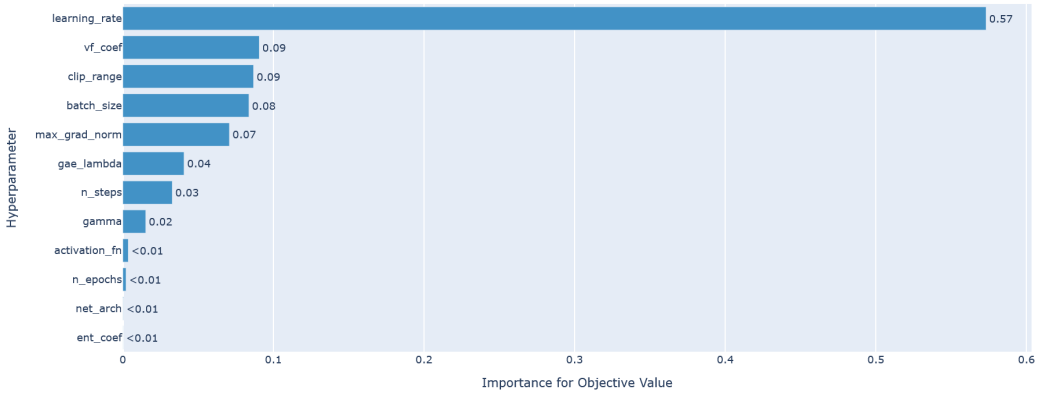


Figure 7.14: Hyperparameter importances evaluated during Optuna study using the FanovaImportanceEvaluator [65].

Table 7.8 provides an overview of Stable Baselines3's [111] default hyperparameters for the RecurrentPPO model with MlpLstm policy and the optimal hyperparameters suggested by the Optuna study.

While some of the optimized hyperparameters are close to the default values, some others show significant differences. The *learning_rate* changed from 3e-4 to ∼2e-4 and

Table 7.8: Comparison of Stable Baselines3's [111] default hyperparameters and Optuna's [3] tuned hyperparameters for the RecurrentPPO model with LSTM policy.

Parameters	Defaults	Optuna Results
learning_rate	3e-4	0.000212299
vf_coef	0.5	0.02879722
clip_range	0.2	0.4
batch_size	128	64
max_grad_norm	0.5	0.6
gae_lambda	0.95	0.99
n_steps	128	256
gamma	0.99	0.9999
activation_function	tanh	tanh
n_epochs	10	1
net_arch	None	small
ent_coef	0.0	3.68E-08

for the *vf_coef*, which represents the coefficient for the value function in the overall loss calculation, Optuna suggests a value ∼0.029 instead of the default 0.5, which favors explorative policy updates. With a *clip_range* of 0.4 instead of 0.2, Optuna proposes less conservative updates and therefore larger changes in the policy. Regarding the *batch_size*, Optuna favors a smaller mini-batch size of 64 instead of the default 128. *max_grad_norm* increased from 0.5 to 0.6 and the *gae_lambda* from 0.95 to 0.99. Optuna suggests to change *n_steps* from 128 to 256 and therefore increase the amount of steps to run per update. *gamma* changed from 0.99 to 0.9999, the *activation_function* is *tanh* in the optimized as well as in the default setting. Instead of using 10 epochs when optimizing the surrogate loss, the optimized parameter for *n_epochs* is 1, which might generally lead to faster training updates and less stable learning. Regarding the specifications of the policy and the value networks, instead of the default *None*, meaning a linear network, the optimized hyperparameter is *small*, which stands for $\text{dict}(\text{pi} = [64, 64], \text{vf} = [64, 64])$ and therefore a two-layer neural network with 64 neurons in each layer for the policy network (pi) as well as for the value function network (vf). The *ent_coef* changed from 0 to ∼4e-8.

In the following simulations, the optimized hyperparameters determined with Optuna are used instead of Stable Baselines3's [111] default hyperparameters for the RecurrentPPO model with MlpLstm policy.

Sigmoid Observation Function, Non-differential Reward Function with ACQ Correction and Tuned Hyperparameters

The following simulations have been conducted with the same simulation setup as specified in Table 7.7, but with the difference of now using tuned instead of default hyperparameters

for the RecurrentPPO (see Table 7.9 for setup overview).

Table 7.9: Setup and results for the simulations discussed in the present subsection.

Setup	
Patient model	RW
Observation function	sigmoid
Reward function	ACQcorrectedNonDifferentialReward
RecurrentPPO hyperparameters	tuned
Results	
[1,0,0]-fraction	1.0
all-[1,0,0]-fraction	1.0
Fitting parameters	R_{lim}= 71.85 τ= 1537.46 R^2 = 0.9332

Figure 7.15 shows the mean and fitted learning curve with standard deviation averaged over the 20 simulations for this paragraph's simulation setup and a baseline showing the average reward over 10000 patients when applying the [1,0,0]-protocol. Within ∼2000 training steps the level of the [1,0,0]-baseline is reached, where the learning curve remains throughout the whole training process.

Using tuned hyperparameters for the training process, reward function ACQcorrectedNonDifferentialReward achieved a [1,0,0]-fraction of 1.0 (as opposed to 0.8363 with default hyperparameters) and an all-[1,0,0]-fraction of 1.0 (as opposed to 0.6223 with default hyperparameters). This means all of the 20 models have learned the desired ET protocol with all actions being CS+-only presentations (see Figure 7.16).

Due to the well-behaving learning curve (see Figure 7.15) and the high values for the [1,0,0]- and all-[1,0,0]-fraction, based on Section 5.5 an exponential curve has been fitted to the mean learning curve of the 20 simulations. The resulting parameters from the fitting process are $R_{\text{lim}} = 71.85$, $\tau = 1537.46$ and $R^2 = 0.9332$ (see also Table 7.9). Based on Equation (5.39) the number of patients (corresponding to RL episodes) required for reaching asymptotic performance is $\frac{5\tau}{\text{extinction trials}} = \frac{5*1537.46}{24} \sim 321$ patients.

Sigmoid Observation Function, Differential Reward Function with ACQ Correction and Tuned Hyperparameters

Similar to reward function ACQcorrectedNonDifferentialReward (see Equation (5.30)), reward function ACQcorrectedDifferentialReward (see Equation (5.31)) is offering an acquisition correction, but has an additional differential element by considering not only the CRs to CS+, but also the CRs to CS-. The resulting reward expresses the percentage of fear extinguished after stage 2 compared to the fear at the end of stage 1 and relative to the fear for CS-. As mentioned in Section 5.3, reward function ACQcorrectedDifferentialReward is inspired by Table 1, index 10 and 13 of

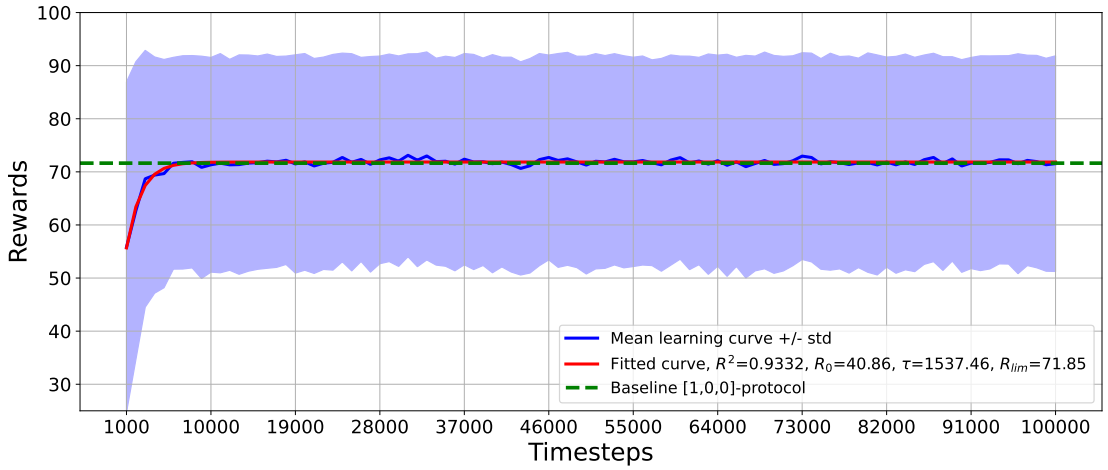


Figure 7.15: Mean and fitted learning curve of 20 simulations with the RW model, sigmoid observation function, reward function ACQCORRECTEDNONDIFFERENTIALREWARD (see Equation (5.30)), and tuned hyperparameters for the RecurrentPPO model. Shown are the mean rewards of the evaluation callbacks each 1000 training steps for 100 episodes averaged over the 20 simulations (bold blue line), the standard deviation (light blue area), the fitted curve (red line), and a baseline showing the average reward over 10000 patients when applying the [1,0,0]-protocol (green dashed line). Within ∼2000 training steps the level of the [1,0,0]-baseline is reached, where the learning curve remains throughout the whole training process.

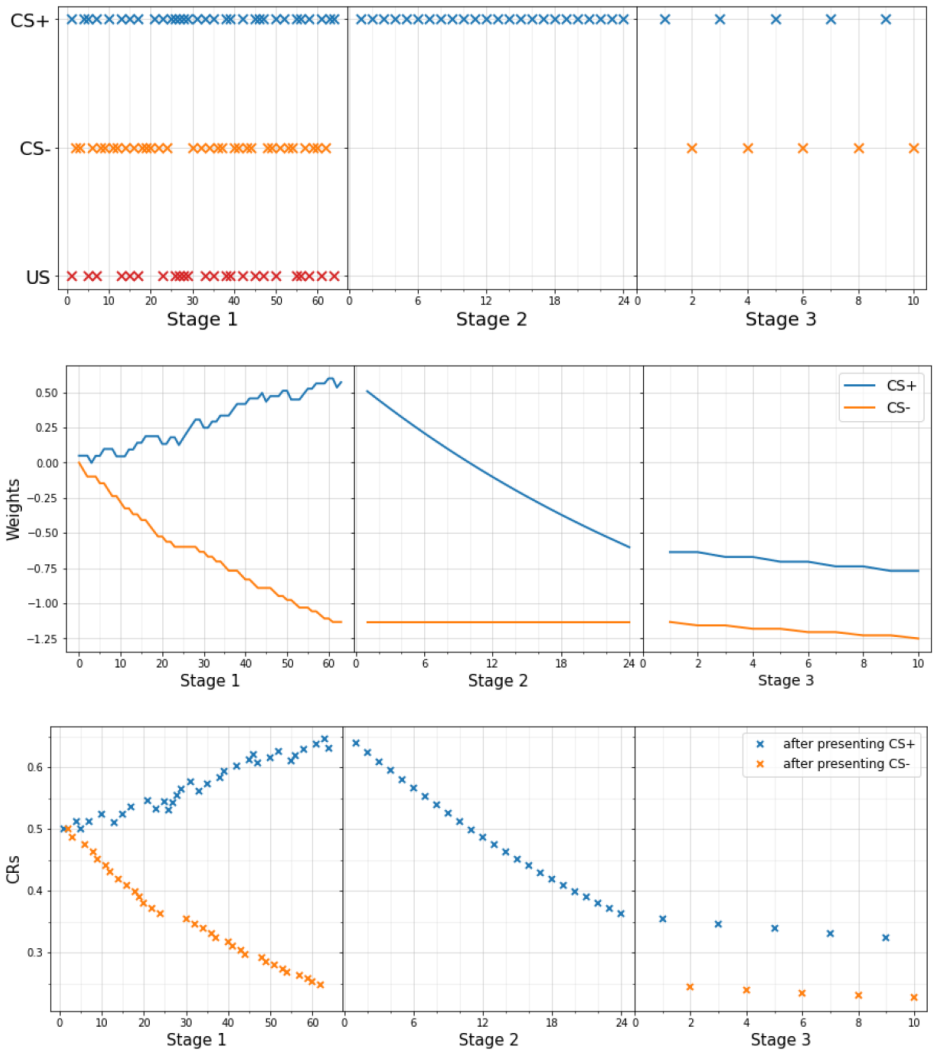


Figure 7.16: ET protocol suggested by all of the RL models trained with the RW model, the sigmoid observation function, reward function ACQcorrectedNonDifferentialReward (see Equation (5.30)), and tuned hyperparameters. Time courses of presented stimuli, evolution of weights and evolution of CRs are shown using an RW learning rate of $\alpha = 0.1$. The ET protocol consists of [1,0,0]-actions only, which leads to decreasing weights for CS+ while keeping CS-'s weights constant and which is the aimed-for ET protocol for the RW patient model.

Lonsdorf et al.'s work from 2019 [83]. Formulating it similarly to Lonsdorf et al, the numerator can get negative if the CRs to CS- are increased during extinction (stage 2) and the rewards can therefore become >100% due to high CRs to CS- in RET (stage 3). This formulation motivates the RL agent to increase the weights and therefore the CRs to CS-. With this strategy high rewards can be achieved, while instead of diminishing fear for CS+, additionally a negative association towards CS- is formed.

When using reward function ACQcorrectedDifferentialReward (see Equation (5.31)) with the RW patient model, the sigmoid observation function, and tuned hyperparameters (see Table 7.8), all models find the discussed ET protocol which maximizes the weights of CS-. The according ET protocol is all actions being [0,1,0] (see Figure 7.18) and the [1,0,0]-fraction as well as the all-[1,0,0]-fraction are therefore 0.

Table 7.10: Setup and results for the simulations discussed in the present subsection.

Setup	
Patient model	RW
Observation function	sigmoid
Reward function	ACQcorrectedDifferentialReward
RecurrentPPO hyperparameters	tuned
Results	
[1,0,0]-fraction	0
all-[1,0,0]-fraction	0

The mean learning curve of the 20 models is shown in Figure 7.17 with values significantly higher than the [1,0,0]-baseline, while not reducing CS+'s weights.

Due to the documented issues, the reward function has been reformulated incorporating a norm expression instead of the subtractions in numerator and denominator, which is discussed in the following subsection.

Sigmoid Observation Function, Reward Function with ACQ Correction Considering Norm of Both CSs' CRs and Tuned Hyperparameters

Using reward function ACQcorrectedNormReward (see Equation (5.32)) includes an ACQ correction that also considers CRs to CS- while avoiding the issues documented for reward function ACQcorrectedDifferentialReward in the last subsection. Instead of subtracting the mean of the CRs to the first $n = 3$ CS+ and those to the first $n = 3$ CS- for the RET in the numerator and for the ACQ in the denominator, the euclidean norm is calculated from those means (see Equation (5.32)).

The following simulations have been conducted using reward function ACQcorrectedNormReward (Equation (5.32)) with the RW patient model, the sigmoid observation function, and tuned hyperparameters (see Table 7.11 for setup overview).

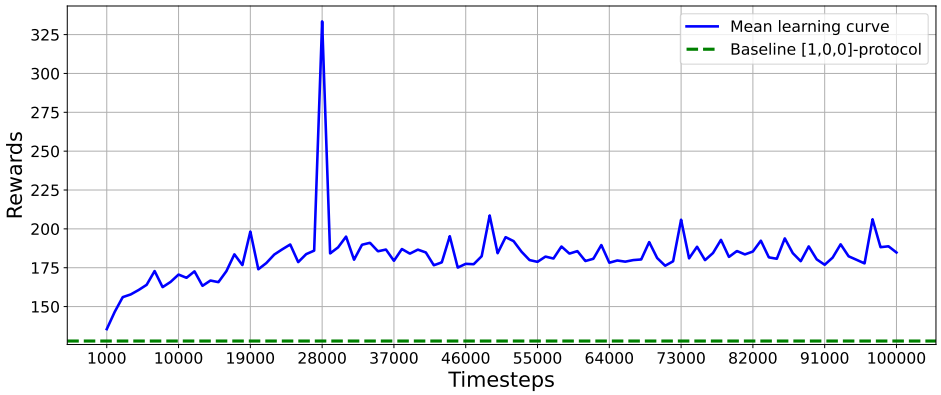


Figure 7.17: Mean learning curve of 20 simulations with the RW model, sigmoid observation function, reward function ACQcorrectedDifferentialReward (see Equation (5.31)), and tuned hyperparameters for the RecurrentPPO model. Shown are the mean rewards of the evaluation callbacks each 1000 training steps for 100 episodes averaged over the 20 simulations and a baseline showing the average reward over 10000 patients when applying the [1,0,0]-protocol (green dashed line). The mean learning curve highly exceeds the [1,0,0]-baseline while not reducing CS+'s weights, due to the exploitation of the reward function's properties.

Table 7.11: Setup and results for the simulations discussed in the present subsection.

Setup	
Patient model	RW
Observation function	sigmoid
Reward function	ACQcorrectedNormReward
RecurrentPPO hyperparameters	tuned
Results	
[1,0,0]-fraction	1.0
all-[1,0,0]-fraction	1.0
Fitting parameters	$R_{\text{lim}} = 67.53$ $\tau = 7943.83$ $R^2 = 0.8363$

The usage of reward function ACQcorrectedNormReward prevents the RL agent from trying to maximize CS-'s CRs and instead leads to desired [1,0,0]- and all-[1,0,0]-fractions of 1.0, meaning that all of the 20 RL models have learned the desired ET protocol of all actions being CS+-only presentations (same evolution of weights and CRs as in Figure 7.16).

The learning curve is well behaving (see Figure 7.19) and the parameters from the

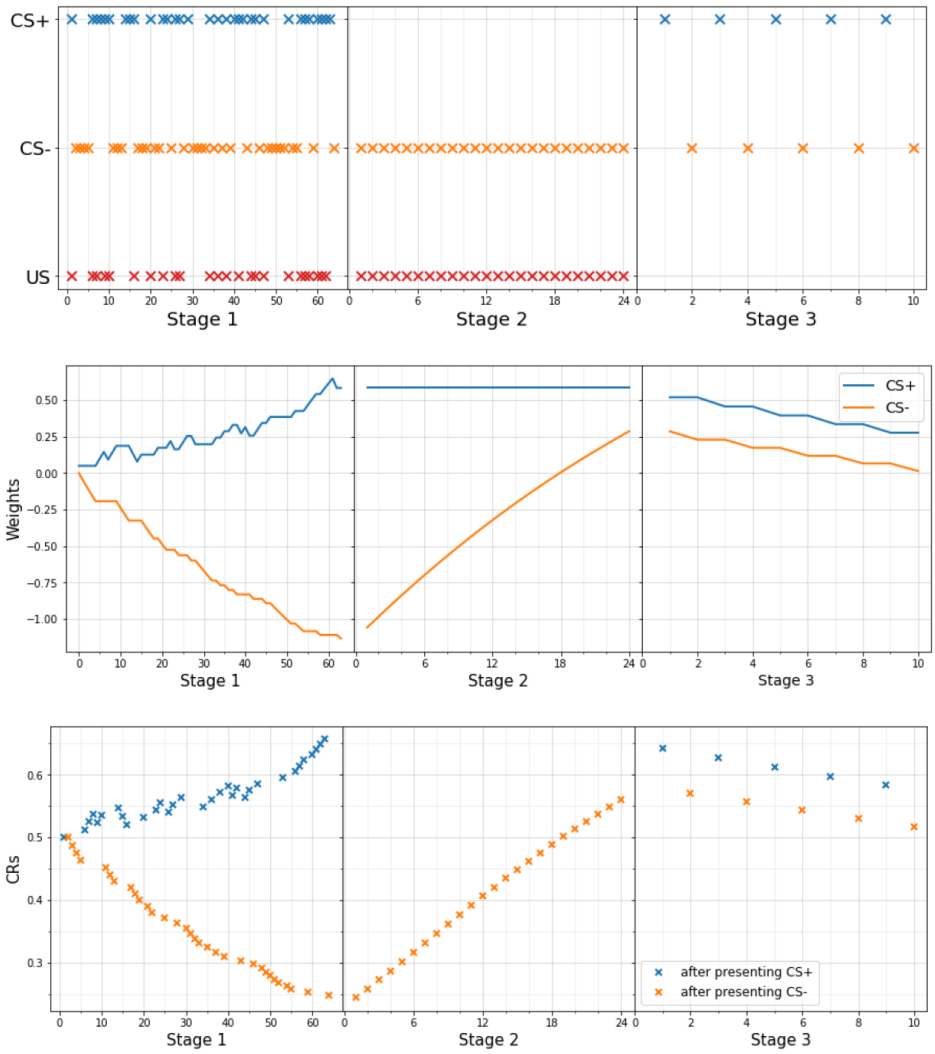


Figure 7.18: ET protocol suggested by all RL models trained with the RW model, the sigmoid observation function, reward function ACQCORRECTEDDIFFERENTIALREWARD (see Equation (5.31)), and tuned hyperparameters. Time courses of presented stimuli, evolution of weights and evolution of CRs are shown using an RW learning rate of $\alpha = 0.1$. The ET protocol consists of [0,1,1]-actions only, which leads to increasing weights for CS- while keeping CS+'s weights constant.

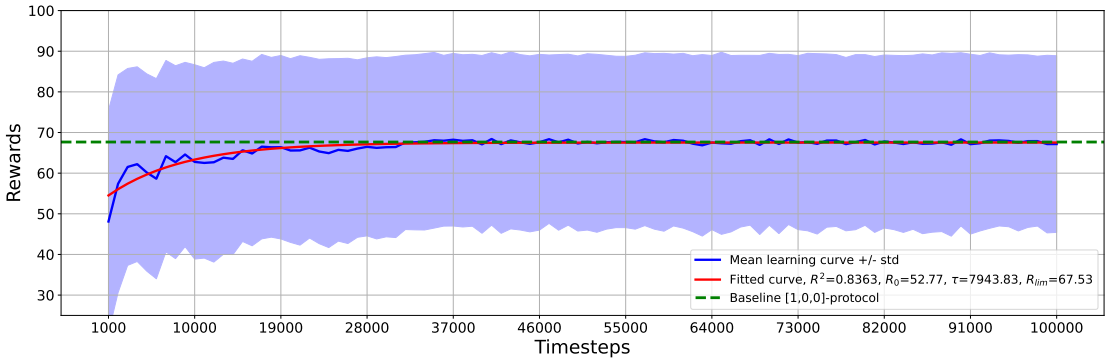


Figure 7.19: Mean and fitted learning curve of 20 simulations with the RW model, sigmoid observation function, reward function ACQCORRECTEDNORMREWARD (see Equation (5.32)), and tuned hyperparameters for the RecurrentPPO model. Shown are the mean rewards of the evaluation callbacks each 1000 training steps for 100 episodes averaged over the 20 simulations (bold blue line), the standard deviation (light blue area), the fitted curve (red line), and a baseline showing the average reward over 10000 patients when applying the [1,0,0]-protocol (green dashed line). The learning curve continues to increase and eventually reaches the level of the [1,0,0]-baseline, where it remains for the rest of the training.

curve fit are $R_{\text{lim}} = 67.53$, $\tau = 7943.83$ and $R^2 = 0.8363$ (see also Table 7.9). Based on Equation (5.39) the number of patients (corresponding to RL episodes) required for reaching asymptotic performance is $\frac{5\tau}{\text{extinction trials}} = \frac{5*7943.83}{24} \sim 1655$ patients. The number of required patients is increased compared to the 321 required patients when using reward function ACQCORRECTEDNONDIFFERENTIALREWARD, which might be due to the slightly more complex formulation of the reward function, making it harder for the RL agent to comprehend the influence of the individual CSs' CRs on the reward.

7.3 Transferability to Other Patient Models (SQ5)

The proceeding simulations focus on answering SQ5, by analyzing whether parameter settings that led to an optimized ET protocol for the RW patient model achieve similarly good results when transferred to the KRW and the DP-KRW model.

For the RW model, the best results were achieved with

- linear observation function
  reward function ABSSUMMEDRETBASEDREWARD (see Equation (5.29))
  default RecurrentPPO model hyperparameters
  (see Table 7.6)

- sigmoid observation function
  reward function ACQCORRECTEDNONDIFFERENTIALREWARD (see Equation (5.30))

tuned RecurrentPPO model hyperparameters
(see Table 7.9)

- sigmoid observation function
  reward function ACQcorrectedNormReward (see Equation (5.32))
  tuned RecurrentPPO model hyperparameters
  (see Table 7.11)

The goal now is to choose a setup used for simulations with the KRW and the DP-KRW model in order to evaluate transferability of (hyper-)parameter settings from one patient model to another.

Regarding the observation function, good results were achieved with both, the linear and the sigmoid function. Considering its underlying properties, it is preferable to choose the sigmoid observation function, as it constrains the CRs to be positive, which is most often the case for the measures typically collected in empirical fear extinction experiments. Another advantage of using the sigmoid observation function is that reducing a particular weight is only profitable up to a point while with a linear mapping there is no diminishing utility in trying to make the weight smaller.

Regarding the reward function, good results were achieved with absSummedRETbasedReward (see Equation (5.29)), ACQcorrectedNonDifferentialReward (see Equation (5.30)), and ACQcorrectedNormReward (see Equation (5.32)). The decision was made in favor of reward function ACQcorrectedNormReward due to the arguably most sophisticated approach offering a correction of the acquisition phase and considering both, CS+ and CS-.

The following subsections outline the results achieved with the sigmoid observation function and reward function ACQcorrectedNormReward (see Equation (5.32)) using the KRW and the DP-KRW patient models. The simulations for both patient models have been conducted twice, once with default and once with tuned hyperparameters for the RecurrentPPO model (see Table 7.8).

7.3.1 Kalman Rescorla-Wagner Model

The Kalman Rescorla-Wagner (KRW) model parameters are described in Section 5.2.2, while based on Kruschke et al. [78] the simulations have been conducted with w_0=0, $\log(\sigma_w^2)$=0.0 and $\log(\sigma_r^2)$=1.3863 (see also Table 6.2).

For the parameter $\log(\tau^2)$, several values have been tested: $\log(\tau^2) \in [-4.60517, -2, -1, 0, 1]$, using the sigmoid observation function and reward function ACQcorrectedNormReward (see Equation (5.32)). The following results were achieved

- $\log(\tau^2)$=-4.60517: maximum rewards of ∼25%
- $\log(\tau^2)$=-2: maximum rewards of ∼60%
- $\log(\tau^2)$=-1: maximum rewards of ∼70%
- $\log(\tau^2)$=0: maximum rewards of $>$ 80%

- $\log(\tau^2)$=1: maximum rewards of $> 80\%$

For better comparison of the proceeding KRW patient model simulations with the previous RW model simulations, $\log(\tau^2)$ has been chosen in a way that leads to a comparable effective learning rate as in the RW patient model. $\log(\tau^2) = -1$ achieved the most comparable maximum rewards of $\sim$70% and is therefore the choice for proceeding simulations.

In the following simulations the KRW patient model with the sigmoid observation function and reward function ACQcorrectedNormReward (see Equation (5.32)) have been used (see Table 7.12 for setup overview). 20 models have been trained for 100000 steps, once with default and once with tuned hyperparameters for the RecurrentPPO model (see Table 7.8 for tuned hyperparameter values). In both cases [1,0,0]-fractions and all-[1,0,0]-fractions of 1.0 were achieved.

Table 7.12: Setup and results for the simulations using the KRW patient model.

Setup	
Patient model	KRW
Observation function	sigmoid
Reward function	ACQcorrectedNormReward
RecurrentPPO hyperparameters	default, tuned
Results	
[1,0,0]-fraction	1.0 (default and tuned hyperparams)
all-[1,0,0]-fraction	1.0 (default and tuned hyperparams)
Fitting parameters with default hyperparams	R_{lim}=73.12 τ=3052.15 R^2=0.9894
Fitting parameters with tuned hyperparams	R_{lim}=73.03 τ=2115.29 R^2=0.8703

Figure 7.20 shows the mean and fitted learning curve with standard deviation averaged over the 20 simulations for this paragraph's simulation setup with default and tuned hyperparameters. In both cases, within $\sim$10000 training steps the level of the [1,0,0]-baseline is reached, where the learning curve remains throughout the whole training process. Due to the fact that in the KRW model the weight updates are based on the Kalman gain and not on learning rates drawn from a prior, the learning curves regulate more quickly with lower standard deviation compared to using the RW patient model.

Using default hyperparameters, the curve fitting parameters are R_{lim}=73.12, τ=3052.15 and R^2=0.9894 (see Figure 7.20a), leading to $\frac{5\tau}{\text{extinction trials}} = \frac{5*3052.15}{24} \sim 636$ required patients (RL episodes) for reaching asymptotic performance (see Equation (5.39)). Using tuned hyperparameters, the curve fitting parameters are R_{lim}=73.03, τ=2115.29,

R^2=0.8703 (see Figure 7.20b), leading to $\frac{5\tau}{\text{extinction trials}} = \frac{5*2115.29}{24} \sim 441$ required patients for reaching asymptotic performance.

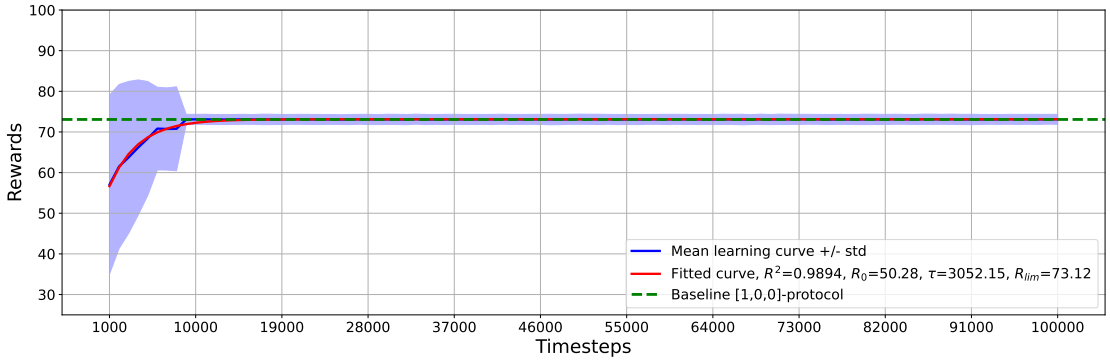
Figure 7.21 shows the optimized ET protocol found by all models, trained with default and tuned hyperparameters using the discussed setup (see Table 7.12). As the [1,0,0]- and all-[1,0,0]-fractions of both 1.0 suggest, it is the [1,0,0]-protocol with all actions being CS+-only presentations. While the [1,0,0]-protocol was the aimed-for ET protocol for the RW patient model, this must not necessarily be the most effective protocol when using the KRW model. Applying the [1,0,0]-protocol still has the desirable outcome of high rewards with extinction of negative associations towards CS+.

Summarizing this section's results, the optimized experimental settings from the simulations with the RW patient model could be successfully transferred to the KRW patient model.

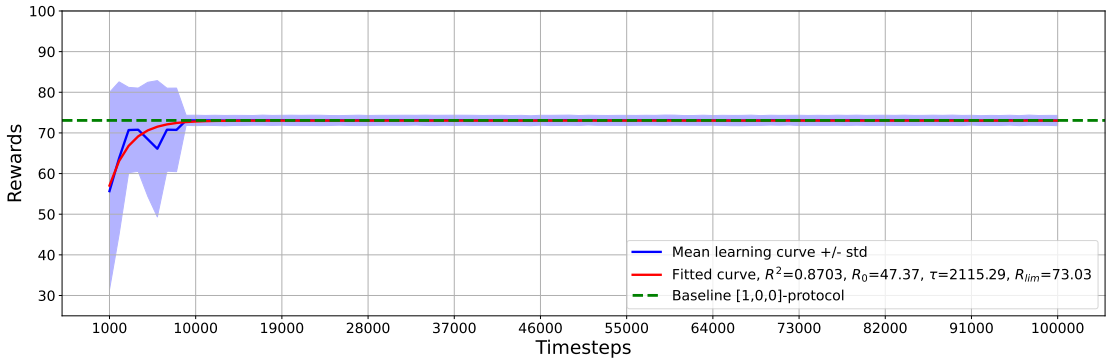
7.3.2 Dirichlet Process-Kalman Rescorla-Wagner Model

The description of the Dirichlet Process-Kalman Rescorla-Wagner (DP-KRW) model parameters is outlined in Section 5.2.3. The following simulations have been conducted using w_0=0, $\log(\sigma_w^2)$=0, $\log(\sigma_r^2)$=1.3863 (taken from Kruschke et al. [78]), α_{DP}=0.1, β=0 and $k_{\text{Max}} = 10$ (decision through trial and error as well as discussions with Samuel Gershman, the author of [50]). Due to reasons discussed in the last Section 7.3.1, $\log(\tau^2)$ has been set to -1 (see also Table 6.3).

In the following simulations the DP-KRW patient model with the sigmoid observation function and reward function ACQcorrectedNormReward (see Equation (5.32)) has been used (see Table 7.13 for setup overview). 20 models have been trained for 100000 steps, once with default and once with tuned hyperparameters for the RecurrentPPO model (see Table 7.8 for tuned hyperparameter values).



(a) Mean learning curve of simulations with the KRW patient model, and default hyperparameters for the RecurrentPPO model.



(b) Mean learning curve of simulations with the KRW patient model and tuned hyperparameters for the RecurrentPPO model.

Figure 7.20: Mean and fitted learning curve of 20 simulations with the KRW model, sigmoid observation function, reward function ACQCORRECTEDNORMREWARD (see Equation (5.32)), and (a) default and (b) tuned hyperparameters for the RecurrentPPO model. Shown are the mean rewards of the evaluation callbacks each 1000 training steps for 100 episodes averaged over the 20 simulations (bold blue line), the standard deviation (light blue area), the fitted curve (red line), and a baseline showing the average reward over 10000 patients when applying the [1,0,0]-protocol (green dashed line). Within ∼10000 training steps the level of the [1,0,0]-baseline is reached, where the learning curve remains throughout the whole training process.

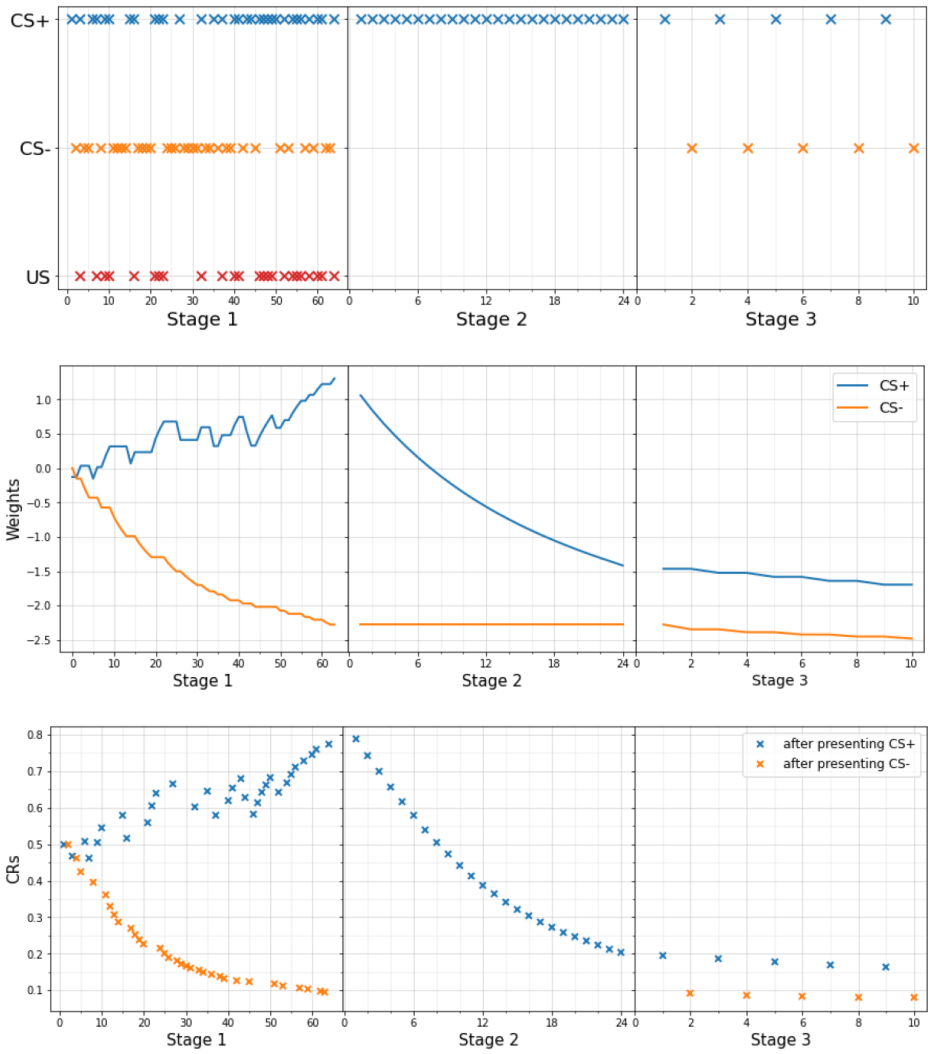


Figure 7.21: ET protocol suggested by all RL models trained with the KRW model, the sigmoid observation function, reward function ACQCORRECTED-NORMREWARD (see Equation (5.32)), default and tuned hyperparameters. Shown are time courses of presented stimuli, evolution of weights and evolution of CRs. The ET protocol consists of [1,0,0]-actions only, which leads to decreasing weights for CS+ while keeping the weights for CS- constant.

Table 7.13: Setup and results for the simulations using the DP-KRW patient model.

Setup	
Patient model	DP-KRW
Observation function	sigmoid
Reward function	ACQcorrectedNormReward
RecurrentPPO hyperparameters	default, tuned
Results	
[1,0,0]-fraction	1.0 (default and tuned hyperparams)
all-[1,0,0]-fraction	1.0 (default and tuned hyperparams)
Fitting parameters with default hpyerparams	R_{lim}=73.09 τ=1306.06 R^2=0.9501
Fitting parameters with tuned hyperparams	R_{lim}=73.09 τ=1023.46 R^2=0.9719

As with the KRW patient model, in both cases [1,0,0]-fractions and all-[1,0,0]-fractions of 1.0 were achieved.

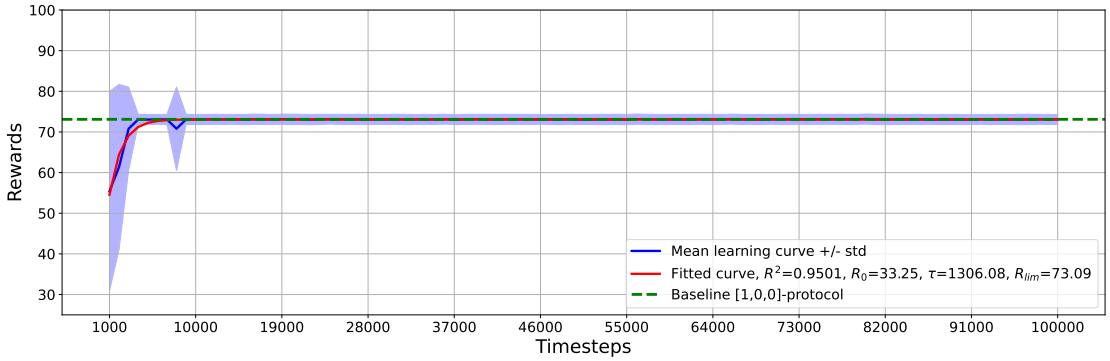
Figure 7.22 shows the mean and fitted learning curve with standard deviation averaged over the 20 simulations for this paragraph's simulation setup with default and tuned hyperparameters. In both cases, within ∼2000 training steps the level of the [1,0,0]-baseline is reached, where the learning curve remains throughout the whole training process. Due to the fact that in the DP-KRW model the weight updates are based on the Kalman gain and not on learning rates drawn from a prior, the learning curves regulate more quickly with lower standard deviation compared to using the RW patient model.

Using default hyperparameters, the curve fitting parameters are R_{lim}=73.09, τ=1306.06 and R^2=0.9501 (see Figure 7.22a), leading to $\frac{5\tau}{\text{extinction trials}} = \frac{5*1306.06}{24} \sim 273$ required patients (RL episodes) for reaching asymptotic performance (see Equation (5.39)). Using tuned hyperparameters, the curve fitting parameters are R_{lim}=73.09, τ=1023.46 and R^2=0.9719 (see Figure 7.22b), leading to $\frac{5\tau}{\text{extinction trials}} = \frac{5*1023.46}{24} \sim 214$ required patients for reaching asymptotic performance.

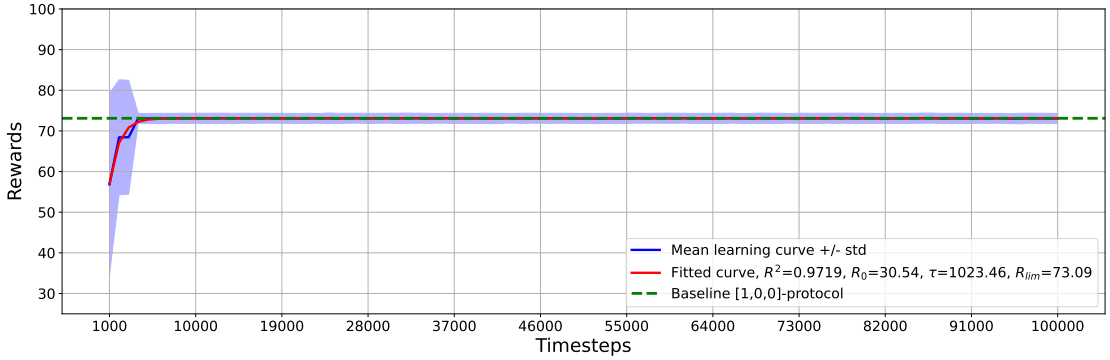
Figure 7.23 shows the optimized ET protocol found by all models, trained with default and with tuned hyperparameters using the discussed setup (see Table 7.13). As the [1,0,0]- and all-[1,0,0]-fractions of 1.0 suggest, it is the [1,0,0]-protocol with all actions being CS+-only presentations. Compared to the previous figures showing the chosen ET protocols, Figure 7.23 offers additional information, as with the DP-KRW model the concept of latent causes has been introduced (see Section 3.4.3). Figure 7.23 shows the evolution of weights for the individual latent causes. The parameter k_{Max} describing the maximum number of latent causes has been set to 10, while only 2 latent causes have been found during the simulations and therefore only the weights of those 2 latent causes have been visualized. Furthermore, the evolution of the posterior probabilities for those 2

latent causes is shown, while essentially throughout all stages latent cause 1 is dominant. Although the [1,0,0]-protocol was the aimed-for ET protocol for the RW model, this must not necessarily be the most effective protocol when using the DP-KRW model. Applying the [1,0,0]-protocol still has the desirable outcome of high rewards with extinction of negative associations towards CS+.

Summarizing this section's results, the optimized experimental settings from the simulations with the RW patient model could be successfully transferred to the DP-KRW patient model.



(a) Mean learning curve of simulations with the DP-KRW patient model, and default hyperparameters for the RecurrentPPO model.



(b) Mean learning curve of simulations with the DP-KRW patient model and tuned hyperparameters for the RecurrentPPO model.

Figure 7.22: Mean and fitted learning curve of 20 simulations with the DP-KRW model, sigmoid observation function, reward function ACQCORRECTEDNORMREWARD (see Equation (5.32)), and (a) default and (b) tuned hyperparameters for the RecurrentPPO model. Shown are the mean rewards of the evaluation callbacks each 1000 training steps for 100 episodes averaged over the 20 simulations (bold blue line), the standard deviation (light blue area), the fitted curve (red line) and a baseline showing the average reward over 10000 patients when applying the [1,0,0]-protocol (green dashed line). Within ∼2000 training steps the level of the [1,0,0]-baseline is reached, where the learning curve remains throughout the whole training process.

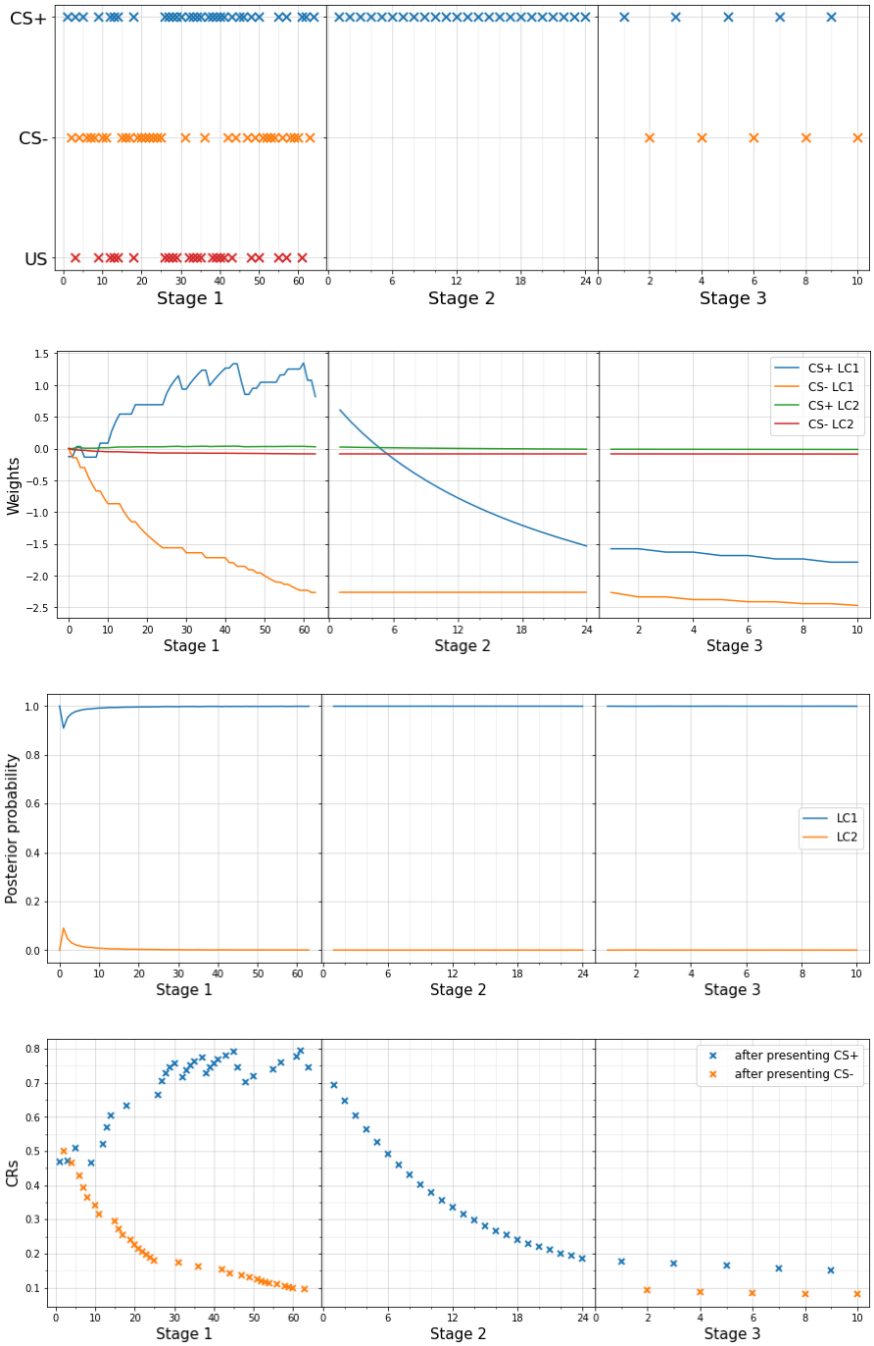


Figure 7.23: ET protocol suggested by all RL models trained with the DP-KRW model, the sigmoid observation function, reward function ACQCORRECTEDNORMREWARD (see Equation (5.32)), and default and tuned hyperparameters. Shown are time courses for presented stimuli, evolution of weights for the latent causes, evolution of posterior probability of the latent causes and evolution of CRs. The ET protocol consists of [1,0,0]-actions only, which leads to decreasing weights for CS+. In all simulations 2 latent causes have been found, while essentially throughout all stages latent cause 1 is dominant.

8 Discussion

The objective of this thesis was to investigate the potential of RL in the optimization of ET protocols. RL-assisted ET might help encounter the lack of general ET design guidelines [17], reduce the problem of therapist scarcity especially in low- and middle-income countries [112], and could be particularly valuable as low-threshold interventions, to treat ADs at a sub-clinical level of symptoms [89].

In the conducted simulations an RL agent served as stand-in for a therapist interacting with a patient, implemented by a computational associative learning model. Goal of the RL algorithm was to maximize the extinction of a previously acquired negative association, with the level of extinction serving as reward.

The thesis' main objective being to investigate the limitations and potential of RL in the optimization of ET protocols, several SQs have been formulated addressing the factors potentially determining the success or failure of using RL approaches in the considered setting:

- SQ1: How many extinction trials lead to robust and efficient ET protocols in the simulations?
- SQ2: Which prior for the RW model learning rate captures empirically reasonable speed of human fear learning and supports finding optimized ET protocols in the simulations?
- SQ3: Which factors to consider in the reward function to help finding an optimized ET protocol?
- SQ4: How does the patient model observation function influence the RL agent's ability to find an optimized ET protocol?
- SQ5: Can parameter settings that lead to an optimized ET protocol for one patient model achieve similarly good results when transferred to other related patient models?

8.1 Conclusions

This thesis' simulations focused on answering SQ1-5, while several conclusions have been drawn from the obtained results. The following listing outlines the most important findings and in parts discusses them in the context of related literature.

- **SQ1** Regarding the number of extinction trials, the results show a clear trend towards simulations using less extinction trials more frequently finding ET protocols that lead to high fear extinction. From the tested numbers of extinction trials $n \in [24, 32, 64]$ the best results could be achieved using 24 trials. This finding is also supported by related studies, like e.g., from Bach et al. [7], who developed a protocol for human FC, where regarding the number of trials in fear extinction training, 96 experts suggested a median of 20 trials as a general guideline. In their 2017 study, Diaz et al. [39] evaluated whether the use of a massive (80 CS presentations) or a moderate (10 CS presentations) number of extinction trials has a different effect on the recovery of extinguished fear, where expectancy was not lowered after massive extinction compared to moderate extinction, if tested in different spatial contexts. The absent effect of massive extinction trials when tested outside of the extinction context, combined with the suggested 20 trials in Bach et al.'s work, point towards the potential of using a more efficient approach with smaller numbers of extinction trials.

- **SQ2** In order to analyze which prior for the Rescorla-Wagner (RW) learning rate α captures empirically reasonable speed of human fear learning while supporting the finding of optimized ET protocols, the priors $\alpha \in [\mathcal{U}(0, 0.2), \mathcal{U}(0, 0.5), \mathcal{U}(0, 1)]$ have been tested. The most efficient [1,0,0]-protocol for the RW model was most frequently recognized when using $\mathcal{U}(0, 0.2)$ and $\mathcal{U}(0, 0.5)$, indicating that smaller learning rates supported the finding of efficient ET protocols in the simulations. The chosen α prior for the proceeding simulations was $\alpha \sim \mathcal{B}(1.5, 3)$, as with this prior smaller values for α are sampled with higher probability, while still considering the possibility of high α values up to $\alpha = 1$. Regarding the empirically reasonable speed of human fear learning, related literature suggests similar α priors: Stussi et al. [128] and Behrens et al. [15, Figure 2e] offer a collection of learning rate estimates from empirical human learning experiments, which lead to a prior with roughly the same parameters as $\alpha \sim \mathcal{B}(1.5, 3)$ when entering mean, median and quantiles from the collection into the web-based probability distribution elicitation tool MATCH Uncertainty Elicitation Tool [94].

- **SQ3** From the tested reward functions, the top results were achieved using reward functions absSummedRETbasedReward (see Equation (5.29)), ACQcorrectedNonDifferentialReward (see Equation (5.30)) and ACQcorrectedNormReward (see Equation (5.32)). Due to the arguably most sophisticated approach offering a correction for ACQ and considering both, CS- and CS+, reward function ACQcorrectedNormReward is the most recommended reward function for future simulations, followed by reward function ACQcorrectedNonDifferentialReward, which also offers a correction for ACQ. Summarizing the results, it has been found that including the CRs to both CSs is favorable compared to only considering CS+ and that in order to find efficient ET protocols, the reward functions have to be designed in a way that they consider properties of the patient model, such that the RL agent is only able to achieve high rewards if

actually extinguishing negative associations.

- **SQ4** Regarding the observation function – depending on other parameter settings – both, the linear and the sigmoid function, have achieved good results. The sigmoid observation function is still preferable over the linear one, as it constrains the CRs to be positive and reducing the CS-'s weight is only profitable up to a point.
- **SQ5** Transferring optimized simulation settings, like the choice on the observation and reward function, and tuned hyperparameters from the RW patient model to the KRW and the DP-KRW model has led to ET protocols successfully extinguishing a previously acquired fear. The optimized parameters could therefore be successfully transferred from one patient model to other related ones.

Summing up, this thesis' results show that parameter settings like the number of extinction trials or the choice on which information to include in the reward function have a significant impact on the agent's ability to find an ET protocol successfully extinguishing a patient's fear and that optimized parameter settings can be transferred from the RW to the KRW and the DP-KRW patient model. However, there also come some limitations with the instructive results of the experiments, which are discussed in the following section.

8.2 Limitations

The first limitation is related to the successive design of the experiments, where individual simulation factors and their influence have not been tested systematically. Instead, individual factors were carried on into the next simulation setup when achieving good results, where new factors were introduced, making it impossible to clearly determine what caused or prevented the success of the experiments. The simulations have been conducted this way, because the focus was primarily on finding a setup achieving optimized ET protocols as opposed to analyzing the effect of individual factors. If the latter was the main point of interest, experiments would have to be conducted in a more structured way, systematically analyzing the individual factors.

A second concern is related to the hyperparameters, which have been tuned using the RW model, the sigmoid observation function and reward function ACQcorrected-NonDifferentialReward (see Equation (5.30)) and have then been transferred to the simulations with other patient models using another reward function. Tuning the recurrentPPO's hyperparameters for each model and parameter setting individually could have led to better results and faster convergence.

8.3 Added Value and Future Outlook

The present and related studies contribute to the application of AI in the treatment of various phobias with the potential of optimizing and personalizing treatment protocols. The knowledge about AI and human fear learning is still growing and the gain of combining

computer-aided learning with human sciences is not yet exhausted. AI-assisted ET has the potential of more efficient, better accessible and personalized treatments, which might be especially beneficial as low-threshold interventions, to treat ADs at a sub-clinical level of symptoms.

The results of this thesis can be seen as foundation for related future work, serving as guideline for factors determining success or failure of RL in optimizing ET protocols, like RL environmental settings, different computational associative learning models as stand-in for the patients, patient model observations functions and reward functions. One possible future extension of this work is training the RL agent on real world patient data, while first experiments can be done using the data from simulated patients to assess the amount of real patient data needed. Another possible approach is sim2sim testing, where an RL model is trained on one and evaluated on another patient model. Related future projects could also include model-based RL approaches or utilize real-time physiological measurements establishing biofeedback-augmented ET.

The long-term perspective of this and related future studies is to support real life phobia treatments, with the potential of increasing therapy availability while reducing costs, offering general guidance in the design of ET therapy protocols and providing a more effective, personalized therapy approach.

Bibliography

[1] A. A. Abdellatif, N. Mhaisen, Z. Chkirbene, A. Mohamed, A. Erbad, and M. Guizani. Reinforcement learning for intelligent healthcare systems: A comprehensive survey. *arXiv preprint arXiv:2108.04087*, 2021.

[2] J. Achiam. Spinning Up in Deep Reinforcement Learning. 2018.

[3] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[4] V. Arun, V. Prajwal, M. Krishna, B. Arunkumar, S. Padma, and V. Shyam. A boosted machine learning approach for detection of depression. In *2018 IEEE symposium series on computational intelligence (SSCI)*, pages 41–47. IEEE, 2018.

[5] A. P. Association. What is exposure therapy? Accessed March 25, 2024, 2019.

[6] D. Bach and R. Dolan. Knowing how much you don't know: A neural organization of uncertainty estimates. *Nature reviews. Neuroscience*, 13:572–86, 07 2012.

[7] D. R. Bach, J. Sporrer, R. Abend, T. Beckers, J. E. Dunsmoor, M. A. Fullana, M. Gamer, D. G. Gee, A. Hamm, C. A. Hartley, et al. Consensus design of a calibration experiment for human fear conditioning. *Neuroscience & Biobehavioral Reviews*, page 105146, 2023.

[8] J. H. Bak, J. Y. Choi, A. Akrami, I. Witten, and J. W. Pillow. Adaptive optimal training of animal behavior. *Advances in neural information processing systems*, 29, 2016.

[9] B. Bakker. Reinforcement learning with long short-term memory. *Neural Information Processing Systems*, 2002.

[10] O. Bălan, A. Moldoveanu, and M. Leordeanu. A machine learning approach to automatic phobia therapy with virtual reality. *Modern Approaches to Augmentation of Brain Function*, pages 607–636, 2021.

[11] B. Bandelow and S. Michaelis. Epidemiology of anxiety disorders in the 21st century. *Dialogues in Clinical Neuroscience*, 17:327–335, 2015.

[12] B. Bandelow, S. Michaelis, and D. Wedekind. Treatment of anxiety disorders. *Dialogues in clinical neuroscience*, 19(2):93–107, 2017.

[13] B. Bandelow, M. Reitt, C. Röver, S. Michaelis, Y. Görlich, and D. Wedekind. Efficacy of treatments for anxiety disorders: a meta-analysis. *International clinical psychopharmacology*, 30(4):183–192, 2015.

[14] A. Bandura. Self-efficacy: toward a unifying theory of behavioral change. *Psychological review*, 84(2):191, 1977.

[15] T. E. J. Behrens, M. W. Woolrich, M. E. Walton, and M. F. S. Rushworth. Learning the value of information in an uncertain world. *Nature Neuroscience*, 10(9):1214–1221, 2007.

[16] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, editors, *NIPS*, pages 2546–2554, 2011.

[17] J. Boehnlein, L. Altegoer, N. K. Muck, K. Roesmann, R. Redlich, U. Dannlowski, and E. J. Leehr. Factors influencing the success of exposure therapy for specific phobia: A systematic review. *Neuroscience & Biobehavioral Reviews*, 108:796–820, 2020.

[18] E. Bøhn, E. M. Coates, S. Moe, and T. A. Johansen. Deep reinforcement learning attitude control of fixed-wing uavs using proximal policy optimization. In *2019 international conference on unmanned aircraft systems (ICUAS)*, pages 523–533. IEEE, 2019.

[19] A. Bohr and K. Memarzadeh. The rise of artificial intelligence in healthcare applications. In *Artificial Intelligence in healthcare*, pages 25–60. Elsevier, 2020.

[20] D. D. Bourgin, J. C. Peterson, D. Reichman, S. J. Russell, and T. L. Griffiths. Cognitive model priors for predicting human decisions. In *International conference on machine learning*, pages 5133–5141. PMLR, 2019.

[21] M. E. Bouton. Context and behavioral processes in extinction. *Learning Memory*, 11, 2004.

[22] J. C. Britton, T. C. Evans, and M. V. Hernandez. Looking beyond fear and extinction learning: considering novel treatment targets for anxiety. *Current behavioral neuroscience reports*, 1:134–143, 2014.

[23] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[24] R. A. Bryant. Post-traumatic stress disorder: a state-of-the-art review of evidence and challenges. *World psychiatry*, 18(3):259–269, 2019.

[25] J. R. Busemeyer and A. Diederich. *Cognitive modeling*. Sage, 2010.

[26] D. Bzdok and A. Meyer-Lindenberg. Machine learning for precision psychiatry: opportunities and challenges. *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*, 3(3):223–230, 2018.

[27] A. Chalkia, N. Schroyens, L. Leng, N. Vanhasbroeck, A.-K. Zenses, L. Van Oudenhove, and T. Beckers. No persistent attenuation of fear memories in humans: A registered replication of the reactivation-extinction effect. *Cortex*, 129:496–509, 2020.

[28] C. Chen, T. Takahashi, S. Nakagawa, T. Inoue, and I. Kusumi. Reinforcement learning in depression: A review of computational research. *Neuroscience & Biobehavioral Reviews*, 55:247–267, 2015.

[29] S. B. Choi, W. Lee, J.-H. Yoon, J.-U. Won, and D. W. Kim. Ten-year prediction of suicide death using cox regression and machine learning in a nationwide retrospective cohort study in south korea. *Journal of affective disorders*, 231:8–14, 2018.

[30] A. Coronato, M. Naeem, G. De Pietro, and G. Paragliola. Reinforcement learning for intelligent healthcare applications: A survey. *Artificial Intelligence in Medicine*, 109:101964, 2020.

[31] A. C. Courville, N. Daw, and D. Touretzky. Similarity and discrimination in classical conditioning: A latent variable account. *Advances in neural information processing systems*, 17, 2004.

[32] A. C. Courville, N. D. Daw, G. J. Gordon, and D. S. Touretzky. Model unvertaint in classical conditioning. *Advances in Neural Information Processing Systems*, 16, 2003.

[33] A. C. Courville, N. D. Daw, and D. S. Touretzky. Bayesian theories of conditioning in a changing world. *TRENDS in Cognitive Sciences*, 10, 2006.

[34] M. Craske, M. B. Stein, T. C. Eley, M. R. Milad, A. Holmes, R. M. Rapee, and H.-U. Wittchen. Anxiety disorders. *Nature Reviews Disease Primers*, 3, 2017.

[35] M. Craske, M. Treanor, C. C. Conway, T. Zbozinek, and B. Verliet. Maximizing exposure therapy: An inhibitory learning approach. *Behaviour Research and Therapy*, 58:10–23, 2014.

[36] M. G. Craske, D. Hermans, and B. Vervliet. State-of-the-art and future directions for extinction as a translational model for fear and anxiety. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 373, 2018.

[37] M. G. Craske, K. Kircanski, M. Zelikowsky, J. Mystkowski, N. Chowdhury, and A. Baker. Optimizing inhibitory learning during exposure therapy. *Behaviour research and therapy*, 46(1):5–27, 2008.

[38] A. R. Delamater. Experimental extinction in pavlovian conditioning: Behavioural and neuroscience perspectives. *Quarterly Journal of Experimental Psychology Section B*, 57(2):97–132, 2004.

[39] M. C. Diaz, V. E. Quezada, V. M. Navarro, M. A. Laborda, and R. Betancourt. The effect of massive extinction trials on the recovery of human fear conditioning. *Revista Mexicana de Psicología*, 34(1):5–12, 2017.

[40] S. E. Dilsizian and E. L. Siegel. Artificial intelligence in medicine and cardiac imaging: harnessing big data and advanced computing to provide personalized medical diagnosis and treatment. *Current cardiology reports*, 16:1–8, 2014.

[41] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, and K. Mankodiya. Towards fog-driven iot ehealth: Promises and challenges of iot in medicine and healthcare. *Future generation computer systems*, 78:659–676, 2018.

[42] A. C. Fernandes, R. Dutta, S. Velupillai, J. Sanyal, R. Stewart, and D. Chandran. Identifying suicide ideation and suicidal attempts in a psychiatric clinical research database using natural language processing. *Scientific reports*, 8(1):7426, 2018.

[43] E. B. Foa and P. M. Carmen. The efficacy of exposure therapy for anxiety-related disorders and its underlying mechanisms: The case of ocd and ptsd. *Annual review of clinical psychology*, 12:1–28, 2016.

[44] E. B. Foa and M. J. Kozak. Emotional processing of fear: exposure to corrective information. *Psychological bulletin*, 99(1):20, 1986.

[45] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. A sticky hdp-hmm with application to speaker diarization. *The Annals of Applied Statistics*, pages 1020–1056, 2011.

[46] G. O. Gabbard and H. Crisp-Han. The early career psychiatrist and the psychotherapeutic identity. *Academic Psychiatry*, 41:30–34, 2017.

[47] C. Gallistel. Extinction from a rationalist perspective. *Behavioural processes*, 90(1):66–80, 2012.

[48] S. J. Gershman. A unifying probabilistic view of associative learning. *Computational Biology*, 2015.

[49] S. J. Gershman, D. M. Blei, and Y. Niv. Context, learning, and extinction. *Psychological review*, 117(1):197, 2010.

[50] S. J. Gershman, M.-H. Monfils, K. A. Norman, and Y. Niv. The computational nature of memory modification. *eLife*, 2017.

[51] S. J. Gershman and Y. Niv. Exploring a latent cause theory of classical conditioning. *Learning & behavior*, 40:255–268, 2012.

[52] G. Gkotsis, A. Oellrich, S. Velupillai, M. Liakata, T. J. Hubbard, R. J. Dobson, and R. Dutta. Characterisation of mental health conditions in social media using informed deep learning. *Scientific reports*, 7(1):1–11, 2017.

[53] M. G. Gottschalk and K. Domschke. Genetics of generalized anxiety disorder and related traits. *Dialogues in clinical neuroscience*, 19(2):159–168, 2017.

[54] S. Graham, C. Depp, E. E. Lee, C. Nebeker, X. Tu, H.-C. Kim, and D. V. Jeste. Artificial intelligence for mental health and mental illnesses: an overview. *Current psychiatry reports*, 21:1–18, 2019.

[55] C. Grillon and M. Davis. Fear-potentiated startle conditioning in humans: Explicit and contextual cue conditioning following paired versus unpaired training. *Psychophysiology*, 34(4):451–458, 1997.

[56] I. Grondman, L. Busoniu, G. A. Lopes, and R. Babuska. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics, part C (applications and reviews)*, 42(6):1291–1307, 2012.

[57] P. M. Groves and R. F. Thompson. Habituation: a dual-process theory. *Psychological review*, 77(5):419, 1970.

[58] A. Guez, R. D. Vincent, M. Avoli, and J. Pineau. Adaptive treatment of epilepsy via batch-mode reinforcement learning. In *AAAI*, volume 8, pages 1671–1678, 2008.

[59] J. He, S. L. Baxter, J. Xu, J. Xu, X. Zhou, and K. Zhang. The practical implementation of artificial intelligence technologies in medicine. *Nature medicine*, 25(1):30–36, 2019.

[60] D. Hermans, M. G. Craske, S. Mineka, and P. F. Lovibond. Extinction in human fear conditioning. *Biological psychiatry*, 60(4):361–368, 2006.

[61] G. E. Hinton. Learning translation invariant recognition in a massively parallel networks. In *International conference on parallel architectures and languages Europe*, pages 1–13. Springer, 1987.

[62] S. G. Hofmann and J. A. J. Smits. Cognitive-behavioral therapy for adult anxiety disorders: a meta-analysis of randomized placebo-controlled trials. *The Journal of Clinical Psychiatry*, 69:621–632, 2008.

[63] A. Howes, J. P. Jokinen, and A. Oulasvirta. Towards machines that understand people. *AI Magazine*, 44(3):312–327, 2023.

[64] W. Hu. The size and burden of mental disorders and other disorders of the brain in europe 2010. *Eur Neuropsychopharmacol*, 21:655–679, 2011.

[65] F. Hutter, H. Hoos, and K. Leyton-Brown. An efficient approach for assessing hyperparameter importance. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 754–762, Bejing, China, 22–24 Jun 2014. PMLR.

[66] F. Jacobi, M. Höfler, J. Strehle, S. Mack, A. Gerschler, L. Scholl, M. A. Busch, U. Maske, U. Hapke, W. Gaebel, et al. Mental disorders in the general population: Study on the health of adults in germany and the additional module mental health (degs1-mh). *Der Nervenarzt*, 85:77–87, 2014.

[67] F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Y. Wang, Q. Dong, H. Shen, and Y. Wang. Artificial intelligence in healthcare: past, present and future. *Stroke and vascular neurology*, 2(4), 2017.

[68] K. B. Johnson, W.-Q. Wei, D. Weeraratne, M. E. Frisse, K. Misulis, K. Rhee, J. Zhao, and J. L. Snowdon. Precision medicine, ai, and the future of personalized health care. *Clinical and translational science*, 14(1):86–93, 2021.

[69] D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21:345–383, 2001.

[70] D. Jurafsky and J. H. Martin. Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition (2023), (draft), 7 january 2023. *URL https://web. stanford. edu/˜ jurafsky/slp3/ed3book. pdf*, 2017.

[71] S. V. Kalmady, R. Greiner, R. Agrawal, V. Shivakumar, J. C. Narayanaswamy, M. R. Brown, A. J. Greenshaw, S. M. Dursun, and G. Venkatasubramanian. Towards artificial intelligence in mental health by improving schizophrenia prediction with multiple brain parcellation ensemble-learning. *npj Schizophrenia*, 5(1):2, 2019.

[72] J. S. Kaplan and D. F. Tolin. Exposure therapy for anxiety disorders: Theoretical mechanisms of exposure and treatment strategies. *Psychiatric Times*, 28(9):33–37, 2011.

[73] R. C. Kessler, O. Demler, R. G. Frank, M. Olfson, H. A. Pincus, E. E. Walters, P. Wang, K. B. Wells, and A. M. Zaslavsky. Prevalence and treatment of mental disorders, 1990 to 2003. *New England Journal of Medicine*, 352(24):2515–2523, 2005.

[74] R. C. Kessler, M. Petukhova, N. A. Sampson, A. M. Zaslavsky, and H.-U. Wittchen. Twelve-month and lifetime prevalence and lifetime morbid risk of anxiety and mood disorders in the united states. *International journal of methods in psychiatric research*, 21(3):169–184, 2012.

[75] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[76] 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. *Nature Medicine*, 24, 2018.

[77] M. A. Kredlow, L. D. Unger, and M. W. Otto. Harnessing reconsolidation to weaken fear and appetitive memories: A meta-analysis of post-retrieval extinction effects. *Psychological Bulletin*, 142(3):314, 2016.

[78] J. Kruschke. Bayesian approaches to associative learning: From passive to active learning. *Learning behavior*, 36:210–26, 09 2008.

[79] Z. Liu, C. Yao, H. Yu, and T. Wu. Deep reinforcement learning with its application for lung cancer detection in medical internet of things. *Future Generation Computer Systems*, 97:1–9, 2019.

[80] A. G. Loerinc, A. E. Meuret, M. P. Twohig, D. Rosenfield, E. J. Bluett, and M. G. Craske. Response rates for cbt for anxiety disorders: Need for standardized criteria. *Clinical psychology review*, 42:72–82, 2015.

[81] T. B. Lonsdorf, M. M. Menz, M. Andreatta, M. A. Fullana, A. Golkar, J. Haaker, I. Heitland, A. Hermann, M. Kuhn, O. Kruse, S. M. Drexler, A. Meulders, F. Nees, A. Pittig, J. Richter, S. Römer, Y. Shiban, A. Schmitz, B. Straube, V. Bram, J. Wendt, J. M. Baas, and C. J. Merz. Don't fear 'fear conditioning': Methodological considerations for the design and analysis of studies on human fear acquisition, extinction, and return of fear. *Neuroscience and Behavioral Reviews*, 77:247–285, 2017.

[82] T. B. Lonsdorf and C. J. Merz. More than just noise: Inter-individual differences in fear acquisition, extinction and return of fear in humans-biological, experiential, temperamental factors, and methodological pitfalls. *Neuroscience & Biobehavioral Reviews*, 80:703–728, 2017.

[83] T. B. Lonsdorf, C. J. Merz, and M. A. Fullana. Fear extinction retention: is it what we think it is? *Biological Psychiatry*, 85(12):1074–1082, 2019.

[84] 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)*, pages 1414–1417. IEEE, 2013.

[85] R. E. Lubow. Latent inhibition. *Psychological bulletin*, 79(6):398, 1973.

[86] U. Lueken and T. Hahn. Personalized mental health: Artificial intelligence technologies for treatment response prediction in anxiety disorders. In *Personalized Psychiatry*, pages 201–213. Elsevier, 2020.

[87] A. Mahmoudi-Nejad, M. Guzdial, and P. Boulanger. Arachnophobia exposure therapy using experience-driven procedural content generation via reinforcement learning (edpcgrl). *Proceedings of the Seventeenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2021)*, pages 164–171, 2021.

[88] T. V. Maia and M. J. Frank. From reinforcement learning models to psychiatric and neurological disorders. *Nature neuroscience*, 14(2):154–162, 2011.

[89] F. Melinscak. Ai exposure therapist for anxiety disorders. `https://www.fwf.ac.at/en/research-radar/10.55776/ESP133`. Accessed: 2024-04-29.

[90] F. Melinscak and D. R. Bach. Computational optimization of associative learning experiments. *PLoS Computational Biology*, 16(1):e1007593, 2020.

[91] D. D. Miller and E. W. Brown. Artificial intelligence in medical practice: the question to the answer? *The American journal of medicine*, 131(2):129–133, 2018.

[92] R. Miller, R. Barnet, and N. Grahame. Assessment of the rescorla-wagner model. *Psychological bulletin*, 117(3):363—386, May 1995.

[93] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

[94] D. Morris, J. Oakley, and J. Crowe. A web-based tool for eliciting probability distributions from experts. *Environmental Modelling & Software*, 52:1–4, 02 2014.

[95] T. B. Murdoch and A. S. Detsky. The inevitable application of big data to health care. *Jama*, 309(13):1351–1352, 2013.

[96] K. M. Myers and M. Davis. Mechanisms of fear extinction. *Molecular Psychiatry*, 12:120–150, 2007.

[97] D. B. Neill. Using artificial intelligence to improve hospital inpatient care. *IEEE Intelligent Systems*, 28(2):92–95, 2013.

[98] I. Nenadić, M. Dietzek, K. Langbein, H. Sauer, and C. Gaser. Brainage score indicates accelerated brain aging in schizophrenia, but not bipolar disorder. *Psychiatry Research: Neuroimaging*, 266:86–89, 2017.

[99] P. J. Norton and C. P. Esther. A meta-analytic review of adult cognitive-behavioral treatment outcome across the anxiety disorders. *The Journal of nervous and mental disease*, 195:521–531, 2007.

[100] S. Parbhoo, J. Bogojeska, M. Zazzi, V. Roth, and F. Doshi-Velez. Combining kernel and model based learning for hiv therapy selection. *AMIA Summits on Translational Science Proceedings*, 2017:239, 2017.

[101] C.-W. Park, S. W. Seo, N. Kang, B. Ko, B. W. Choi, C. M. Park, D. K. Chang, H. Kim, H. Kim, H. Lee, et al. Artificial intelligence in health care: Current applications and issues. *Journal of Korean medical science*, 35(42), 2020.

[102] V. L. Patel, E. H. Shortliffe, M. Stefanelli, P. Szolovits, M. R. Berthold, R. Bellazzi, and A. Abu-Hanna. The coming of age of artificial intelligence in medicine. *Artificial intelligence in medicine*, 46(1):5–17, 2009.

[103] I. P. Pavlov. *Conditioned reflexes: an investigation of the physiological activity of the cerebral cortex*. Oxford Univ. Press, 1927.

[104] J. M. Pearce and M. E. Bouton. Theories of associative learning in animals. *Annual review of psychology*, 52(1):111–139, 2001.

[105] J. M. Pearce and G. Hall. A model for pavlovian learning: Variations in the effectiveness of conditioned but not of unconditioned stimuli. *Psychological Review*, 87, 1980.

[106] B. K. Petersen, J. Yang, W. S. Grathwohl, C. Cockrell, C. Santiago, G. An, and D. M. Faissol. Precision medicine as a control problem: Using simulation and deep reinforcement learning to discover adaptive, personalized multi-cytokine therapy for sepsis. *arXiv preprint arXiv:1802.10440*, 2018.

[107] A. Pouget, J. Beck, W. Ma, and P. Latham. Probabilistic brains: Knowns and unknowns. *Nature neuroscience*, 16, 08 2013.

[108] M. L. Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.

[109] N. C. Rabinowitz, F. Perbet, H. F. Song, C. Zhang, S. M. A. Eslami, and M. Botvinick. Machine theory of mind. *arXiv preprint arXiv:1802.07740v2*, 2018.

[110] A. Raffin. Rl baselines3 zoo. `https://github.com/DLR-RM/rl-baselines3-zoo`, 2020.

[111] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[112] S. Rathod, N. Pinninti, M. Irfan, P. Gorczynski, P. Rathod, L. Gega, and F. Naeem. Mental health service provision in low-and middle-income countries. *Health services insights*, 10:1178632917694350, 2017.

[113] R. A. Rescorla. Pavlovian conditioned inhibition. *Psychological bulletin*, 72(2):77, 1969.

[114] R. A. Rescorla. Spontaneous recovery. *Learning & Memory*, 11(5):501–509, 2004.

[115] R. A. Rescorla and A. R. Wagner. A theory of pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. *Classical Conditioning II: Current Research and Theory*, 1972.

[116] A. Sau and I. Bhakta. Artificial neural network (ann) model to predict depression among geriatric population at a slum in kolkata, india. *Journal of clinical and diagnostic research: JCDR*, 11(5):VC01, 2017.

[117] M. Schiele and K. Domschke. Epigenetics at the crossroads between genes, environment and resilience in anxiety disorders. *Genes, Brain and Behavior*, 17(3):e12423, 2018.

[118] D. Schiller, M.-H. Monfils, C. M. Raio, D. C. Johnson, J. E. LeDoux, and E. A. Phelps. Preventing the return of fear in humans using reconsolidation update mechanisms. *Nature*, 463(7277):49–53, 2010.

[119] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[120] T. Schürmann and P. Beckerle. Personalizing human-agent interaction through cognitive models. *Frontiers in Psychology*, 11:561510, 2020.

[121] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

[122] D. R. Shanks. Forward and backward blocking in human contingency judgement. *The Quarterly Journal of Experimental Psychology*, 37(1):1–21, 1985.

[123] A. B. Shatte, D. M. Hutchinson, and S. J. Teague. Machine learning in mental health: a scoping review of methods and applications. *Psychological medicine*, 49(9):1426–1448, 2019.

[124] S. M. Shortreed, E. Laber, D. J. Lizotte, T. S. Stroup, J. Pineau, and S. A. Murphy. Informing sequential clinical decision-making through reinforcement learning: an empirical study. *Machine learning*, 84:109–136, 2011.

[125] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

[126] K. S. Springer, H. C. Levy, and D. F. Tolin. Remission in cbt for adult anxiety disorders: a meta-analysis. *Clinical psychology review*, 61:1–8, 2018.

[127] T. G. Stampfl and D. J. Levis. Essentials of implosive therapy: a learning-theory-based psychodynamic behavioral therapy. *Journal of Abnormal Psychology*, 72(6):496, 1967.

[128] Y. Stussi, G. Pourtois, A. Olsson, and D. Sander. Learning biases to angry and happy faces during pavlovian aversive conditioning. *Emotion*, 21:742–756, 03 2020.

[129] S. Su, Y. Wang, W. Jiang, W. Zhao, R. Gao, Y. Wu, J. Tao, Y. Su, J. Zhang, K. Li, et al. Efficacy of artificial intelligence-assisted psychotherapy in patients with anxiety disorders: a prospective, national multicenter randomized controlled trial protocol. *Frontiers in Psychiatry*, 12:799917, 2022.

[130] A. Subramanian, S. Chitlangia, and V. Baths. Reinforcement learning and its connections with neuroscience and psychology. *Neural Networks*, 145:271–287, 2022.

[131] R. S. Sutton and A. G. Barto. Reinforcement learning: An introduction. 2020.

[132] C.-Y. Tang, C.-H. Liu, W.-K. Chen, and S. D. You. Implementing action mask in proximal policy optimization (ppo) algorithm. *ICT Express*, 6(3):200–203, 2020.

[133] J. G. Trafton, L. M. Hiatt, B. Brumback, and J. M. McCurry. Using cognitive models to train big data models with small data. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1413–1421, 2020.

[134] L. J. Van Hamme and E. A. Wasserman. Cue competition in causality judgments: The role of nonpresentation of compound stimulus elements. *Learning and Motivation*, 25(2):127–151, 1994.

[135] R. Vincent. *Reinforcement learning in models of adaptive medical treatment strategies*. McGill University (Canada), 2014.

[136] A. R. Wagner and R. A. Rescorla. Inhibition in pavlovian conditioning: Application of a theory. *Inhibition and learning*, pages 301–336, 1972.

[137] S. Watanabe. Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance. *arXiv preprint arXiv:2304.11127*, 2023.

[138] B. Widrow and M. E. Hoff. Adaptive switching circuits. In *1960 IRE WESCON Convention Record, Part 4*, pages 96–104, New York, 1960. IRE.

[139] B. Xin, H. Yu, Y. Qin, Q. Tang, Z. Zhu, et al. Exploration entropy for reinforcement learning. *Mathematical Problems in Engineering*, 2020, 2020.

[140] C. Yu, Y. Dong, J. Liu, and G. Ren. Incorporating causal factors into reinforcement learning for dynamic treatment regimes in hiv. *BMC medical informatics and decision making*, 19:19–29, 2019.

[141] C. Yu, J. Liu, S. Nemati, and G. Yin. Reinforcement learning in healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(1):1–36, 2021.

[142] Y. Zhao, M. R. Kosorok, and D. Zeng. Reinforcement learning design for cancer clinical trials. *Statistics in medicine*, 28(26):3294–3315, 2009.

Acronyms

ACQ Acquisition. 28, 36–38, 62, 68, 69, 72, 86

ADs Anxiety Disorders and Anxiety-Related Disorders. iii, v, 1, 2, 6–8, 22–24, 85, 88

AI Artificial Intelligence. v, 1, 2, 22–26, 87, 88

CR Conditioned Response. 8, 9, 14, 27–31, 34–38, 54, 56, 57, 60–64, 66, 69, 71–76, 80, 84, 86, 87

CS Conditioned Stimulus. 8–13, 28, 30, 31, 33, 35–38, 41, 47, 48, 54–64, 66, 69, 71–76, 78, 80–82, 84, 86, 87

DP-KRW Dirichlet Process-Kalman Rescorla-Wagner. v–vii, ix, 5, 10, 13, 28, 33–36, 41, 44, 75, 76, 78, 81–84, 87

DTR Dynamic Treatment Regime. 22, 23

EI Expected Improvement. 20, 21

EL Extinction Learning. v, 1, 2, 6, 8, 9, 27

ET Exposure Therapy. iii, v, viii, ix, 1, 2, 4–9, 14, 22–24, 27–30, 37, 38, 40, 41, 45–48, 52, 54–58, 60, 61, 63, 64, 66, 67, 69, 71–74, 78, 80–82, 84–88

FC Fear Conditioning. v, 6, 8, 9, 27, 29, 30, 42, 86

HIV Human Immunodeficiency Virus. 23

KDE Kernel Density Estimators. 21

KRW Kalman Rescorla-Wagner. v–vii, ix, 5, 10–12, 28, 31–33, 35, 41, 43, 44, 75–81, 87

LDS linear-Gaussian dynamical system. 11, 32

LSTM Long Short-Term Memory. vii, 68

MDP Markov Decision Process. 14, 27

ML Machine Learning. 2, 23–25

MLP Multilayer Perceptron. 16, 39

MlpLstm Multilayer Perceptron Long Short-Term Memory. vii, 16, 27, 38, 39, 67, 68

PPO Proximal Policy Optimization. vii, x, 6, 16, 17, 19, 20, 27, 38, 39, 48, 54, 56, 61, 62, 64, 67–70, 73, 75–79, 83, 87

RET Retention. 36–38, 47, 72

RL Reinforcement Learning. iii–vi, viii, ix, 2–4, 6, 14, 16, 17, 20, 22–30, 35–38, 40–42, 45–48, 52, 54–58, 60–64, 66, 67, 69, 71–75, 77, 80, 81, 84–86, 88

rl_zoo3 RL Baselines3 Zoo. 38, 40, 45, 67

RoF Return of Fear. 9, 27–30, 36, 37, 54, 56, 62

RW Rescorla-Wagner. iii–v, vii–ix, 4, 5, 10–12, 28, 31, 32, 35, 41, 43, 46–48, 53–66, 69–75, 77, 78, 81, 82, 85–87

SGD Stochastic Gradient Descent. 20

SQ Subquestion. vi, 4, 5, 28, 30, 43, 46, 47, 54, 75, 85–87

TPE Tree-Structured Parzen Estimator. v, 20, 45

UR Unconditioned Response. 8

US Unconditioned Stimulus. 8–14, 28, 30, 31, 33, 34, 41, 56, 58

VR Virtual Reality. 7, 24