

# Supporting Task Switching with Reinforcement Learning

Alexander Lingler  
Digital Media Lab  
University of Applied Sciences  
Upper Austria  
Hagenberg, Austria  
alexander.lingler@fh-hagenberg.at

Dinara Talypova  
Digital Media Lab  
University of Applied Sciences  
Upper Austria  
Hagenberg, Austria  
dinara.talypova@fh-hagenberg.at

Jussi P.P. Jokinen  
Cognitive Science  
University of Jyväskylä  
Jyväskylä, Finland  
jussi.p.p.jokinen@jyu.fi

Antti Oulasvirta  
Department of Information and  
Communications Engineering  
Aalto University  
Helsinki, Finland  
antti.oulasvirta@aalto.fi

Philipp Wintersberger  
Digital Media Lab  
University of Applied Sciences  
Upper Austria  
Hagenberg, Austria  
& TU Wien  
philipp.wintersberger@fh-  
hagenberg.at



**Figure 1:** We devised a balancing game to study AI-based assistance that helps users in task switching. The game consists of two platforms where only one can be actively controlled at a time. The goal is to balance the balls as long as possible on top of their platforms. To support users, we implemented a reinforcement learning (RL) based attention management system (AMS) that automatically switches between the platforms (indicated by a sound and a change of the active platform's color). The AMS agent has full access to the states of the subtasks and learns a policy to switch in a way that humans' game scores increase.

## ABSTRACT

Attention management systems aim to mitigate the negative effects of multitasking. However, sophisticated real-time attention management is yet to be developed. We present a novel concept for attention management with reinforcement learning that automatically switches tasks. The system was trained with a user model based on principles of computational rationality. Due to this user model, the system derives a policy that schedules task switches by considering human constraints such as visual limitations and reaction times. We evaluated its capabilities in a challenging dual-task

balancing game. Our results confirm our main hypothesis that an attention management system based on reinforcement learning can significantly improve human performance, compared to humans' self-determined interruption strategy. The system raised the frequency and difficulty of task switches compared to the users while still yielding a lower subjective workload. We conclude by arguing that the concept can be applied to a great variety of multitasking settings.

## CCS CONCEPTS

• **Human-centered computing** → HCI theory, concepts and models; **Interactive systems and tools**; **Ubiquitous computing**; **User interface management systems**; **Empirical studies in HCI**.

## KEYWORDS

Interruption, Notification, Task Switching, Machine Learning, Artifact or System, Lab Study, Quantitative Methods



This work is licensed under a Creative Commons Attribution-NoDerivs International 4.0 License.

CHI '24, May 11–16, 2024, Honolulu, HI, USA  
© 2024 Copyright held by the owner/author(s).  
ACM ISBN 979-8-4007-0330-0/24/05  
<https://doi.org/10.1145/3613904.3642063>

**ACM Reference Format:**

Alexander Lingler, Dinara Talypova, Jussi P.P. Jokinen, Antti Oulasvirta, and Philipp Wintersberger. 2024. Supporting Task Switching with Reinforcement Learning. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11–16, 2024, Honolulu, HI, USA*. ACM, New York, NY, USA, 18 pages. <https://doi.org/10.1145/3613904.3642063>

## 1 INTRODUCTION

Human attention is a limited resource, but it is stressed more than ever before in history [2, 81]. Permanently refocusing attention and alternating between activities and tasks comes with a price, including severe accidents [50] and significant socio-economic costs [2, 76]. Sequential multitasking [70] results in “switch costs”, which are characterized by the time and effort needed to stop one and engage in another task. Additionally, external interruptions increase stress, frustration, and error rates while impeding overall performance [6, 47, 52]. Still, multitasking should not, and cannot, be eliminated. In the future, cooperating with machines/algorithms while engaging in other tasks will be common in various domains such as mobility, manufacturing, or health. Consequently, it was argued that “*managing user attention has emerged as a critical challenge*” [15].

The negative effects of multitasking and task switching have led to calls for “Attentive User Interfaces” (AUIs) or “Attention Management Systems” (AMSs). AMSs are systems that “*computationally seek to balance a user’s need for minimal disruption and the application’s need to efficiently deliver information*” [5, 6] and try to manage user attention appropriately – for example, by precisely timing interruptions [80]. It is known that task interruptions are less detrimental when they happen at task boundaries (i.e., small breaks between subtasks) or when humans are relaxed [6, 37], and the application of AUI concepts have shown to be beneficial in various settings from office work [6, 38, 39] to driving [84, 85]. However, beyond conceptual works and theories, AMSs in real-world settings hardly exist. Various contributions so far have evaluated the psychological elements of task switching [40, 79] or modeling of user behavior [29, 42, 48]. Still, despite the existing theories and concepts, it is unknown how to develop an algorithm or system that integrally cooperates with the user and helps them to multitask better. Many existing approaches hardly scale and suffer from problems such as (1) limited scope, (2) potentially inflexible or incomplete theories and heuristics, which in turn can lead to (3) highly complex architectures [5].

For example, existing concepts have dedicated components to detect potential breakpoints or subtask boundaries, where the underlying data was labeled either by experts [38] or by the users themselves [48]. This makes such solutions hard to develop and difficult to transfer to other settings. In a real-time context, data labeling could be infeasible, since heuristics are restricted to our understanding of what constitutes “good” task switching (which has shown to be quite diverse between people [22]). Cognitive architectures such as ACT-R in turn have issues in predicting “*how task interleaving behavior adapts to changes in uncertainty related to the task states*” [43]. This partly stems from the fact that the cognitive processes (i.e., the true costs of interruptions and users’ adaptations) are hardly observable.

We claim that reinforcement learning (RL) and computational rationality (CR) could have a high potential to overcome these limitations and help to build AMSs that benefit human multitasking for multiple reasons. First, AMSs must succeed in a probabilistic sequential decision problem: they need to continuously observe the process of humans operating on different tasks and derive a good strategy for interruptions (i.e., task switches). Thereby, rewards are sparse – the benefit of appropriate interruptions on users’ performance are not immediately visible, while the negative effects of inappropriate switches accumulate over time. RL can build policies for sequences of actions to account for the issue of delayed rewards. Further, while approaches based on cognitive architectures require scripted policies that become highly complex due to the sheer number of behavioral strategies, the policy of RL agents is learned and only driven by rewards. Second, RL has shown to be a suitable approach for improving human attention, cognition, and behavior in the field of HCI. To provide some examples, existing works have shown that RL can explain human behavior in task switching [29], determine timings for intelligent suggestions [91], support human-AI cooperation in visual search tasks [28], optimize physical activities embedded in just-in-time interventions [51], or to provide adaptive time pressure feedback to improve user performance in arithmetic tasks [88]. Collectively, these studies show that RL can act as an enabler for intelligent interventions that fit in naturally with human activities. Third, RL in combination with computational rationality (CR) can help to better understand users and build better cooperative agents [35]. Consequently, we believe RL can be used to coordinate operator resources (i.e., properly schedule/assign tasks) in a way to optimize toward an objective function. This function could aim at increasing the overall performance of the user or optimize towards other parameters, for example, minimizing physiological stress markers.

**We hypothesize, that an RL-based AMS that schedules tasks for users has the potential to improve their performance in a multitasking setting, compared to a situation where the users themselves make the decision when to attend to a particular task.**

Given the focus on task performance, it is not unthinkable that the algorithm reveals suitable strategies for switching that are unknown or even seem counter-intuitive, ultimately becoming a “superhuman” extension for the user. In this paper, we present a prototypical implementation of this concept and demonstrate its beneficial effect on overall performance in a dual-task setting (see Figure 1). Our contributions are as follows:

- We introduce the concept of RL-based attention management systems that observe the users’ environment(s) and automatically switch between tasks so that a defined reward (in our prototypical case, human performance) increases.
- The concept was investigated on hand of a fast-paced dual-task. Therefore, we implemented a simple game where the player has to balance two balls on two platforms, but only one platform can be controlled at a given time (see Figure 1). The game requires players to switch between tasks in a visual-motoric activity with high temporal demand, representing a

real-time multitasking situation (similar to other skill-based tasks such as driving) that has not been extensively addressed by AMSs so far. The prototype is hosted on a public GitHub repository along with the code and all the utilized RL models and algorithms<sup>1</sup>. We call for using, extending, and improving the tool for researching and designing future AMSs.

- Implementing this concept requires training of the algorithm either on humans (which would demand a lot of resources and could yield frustration due to the random actions of the RL agents during training) or models that closely replicate human behavior. We implemented two types of user models performing the balancing tasks to train the AMS. One model has unlimited access to perceiving the environment and acting instantly. The other model is built upon principles of “computational rationality” and uses “cognitive constraints” [35, 43] to simulate imperfect perception and reaction times.
- The resulting systems were evaluated in a quantitative lab experiment. We compared versions of the AMS that are trained with the two models (an *unconstrained* and a *cognitive* model) to users’ self-chosen task switching strategy (*no supervisor* condition) as well as a version where task switches were just indicated by the AMS and confirmed by the users themselves (*notification* condition). The results, based on a set of performance (game scores) and behavioral measurements data (reaction times, average time on task, etc.), show that the AMS trained with cognitive constraints can significantly improve human performance in the dual-task setting, confirming the main hypothesis of this work.
- We provide a detailed but critical outlook and call for researching the approach in other multitasking situations while experimenting with additional parameters, constraints, and reward functions.

## 2 RELATED WORK

### 2.1 Multitasking Theory

Research on multitasking, i.e., “*performing two or more activities in a short period of time*” [83], has a long history. Salvucci et al. [70] presented the “multitasking continuum”, arguing that sequential and concurrent multitasking represent two sides of a spectrum rather than being different processes. While in concurrent multitasking “*tasks are, in essence, performed at the same time*”, task switching is characterized by longer phases dedicated attention towards activities [70], see Figure 2. Task switching and interruptions have been extensively addressed in past research [9, 12, 13, 16, 32, 34, 52, 66]. Interruptions can be defined as “*introduction to a new task or tasks on top of the ongoing activity, often unexpectedly, resulting in conflicts and loss of attention on the current activity, failing to resume the work where it was interrupted*” [5, 58], which requires humans to rehearse and clean the problem state. Consistently, studies have demonstrated that interruptions can disrupt cognitive processes, leading to performance decrease, as well as increased stress and error rates [6, 74, 87]. Interruption costs are influenced by factors such as task complexity, interruption frequency, duration, or the nature

of the secondary task [13, 16, 32]. Resumption lags — the time taken to return to the before-interrupted task — also impact performance. The transitions require cognitive resources for task switching and reactivation of relevant information (recall, see Figure 2).

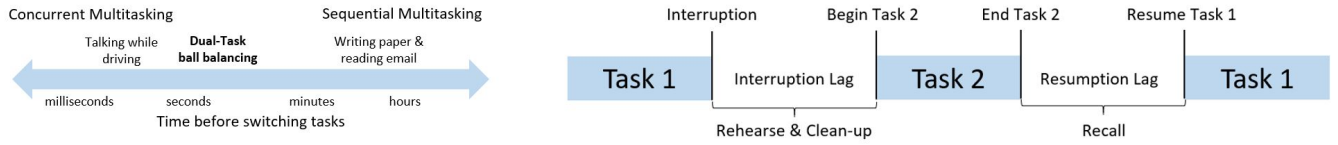
Theories such as the Threaded Cognition Theory [11] try to offer insights into these interruption-related phenomena. According to this theory, cognitive processes are not isolated but interconnected threads that can operate in parallel [70]. Resources execute processes exclusively in service of one task thread at a time. Interruptions can disrupt these threads, resulting in conflicts for cognitive resources. The theory suggests that the cognitive system dynamically allocates resources to various threads, adapting to task demands and priorities [69]. When an interruption occurs, the cognitive system must shift resources from the primary to the secondary task thread. Resuming the primary task requires the re-allocation of resources back to the original thread, which involves cognitive effort and potential delays. Longer interruption- and resumption lags can lead to interference between threads, resulting in performance decrements. The precise timing of interruptions, for example, between subtasks or in times of low mental workload [6], is a potential solution to mitigate the negative effects of interruption- and resumption lags. Multiple studies revealed that it is better to be interrupted between (sub)tasks than in the middle of a task [1, 36, 67, 85]. One of the explanations comes from the memory-for-problem-states theory, suggesting that individuals are more likely to have an active problem-solving state during the middle of a task, but not between distinct tasks [13].

### 2.2 Attention Management Systems

A core idea of AMSs is to strategically time notifications to minimize interruption costs, enhance overall task performance, and provide users with a less stressful environment [5, 6]. Calls for AMSs range back to the early days of ubiquitous computing [5, 82]. Horvitz [33] suggested to “*consider the status of a user’s attention in the timing of services [...] while considering costs and benefits of deferring actions*”, and also recent guidelines for human-AI interaction argue to “*time services based on context*” [4]. Regarding potential features, researchers have proposed to (1) find better timings for interruptions based on interruptibility and (2) adapt user interfaces in a way to ease task resumption. A large body of work has looked into gauging user interruptibility [93] and detecting opportune moments for notification delivery [24, 39, 63] by using sensors and machine-learning models. Subsequent mitigation strategies include not only interruption timing but also an indication of availability status [92] or the use of aids to help users resume their tasks more effectively after an interruption occurs [75]. Considering task switching support, it has been shown that properly organized interfaces can also ease task resumption [60] by providing cues. Task resumption cues can be designed in an explicit (providing specific information about the task) or implicit (guiding users’ attention, for example, their gaze) way, and they can be communicated before, during, and after interruptions using different modalities [72].

However, sophisticated real-time AMSs hardly exists. So far, proposed concepts build upon computational models or perform a data-driven approach [5]. Systems using computational models conduct decisions by predicting the behavior of users. Therefore, Brumby

<sup>1</sup><https://github.com/reapphil/AltentiveMultitasking>



**Figure 2: The multitasking continuum describes the relationship between concurrent and sequential multitasking. The balancing task used for this project contains aspects of both concurrent and sequential multitasking (adapted from Salvucci et al. [70] left). Task switching results in switch costs given by the interruption and the resumption lags (right).**

et al. [14] distinguish between cognitive architectures (such as ACT-R), constraint modeling, and uncertainty modeling. A problem with systems that build upon computational multitasking models is that they “*focus on a different grain size of behavior and/or different tradeoffs of aggregate behavior vs. individual behavior*” and are not always flexible enough to cover the uncertainties and individual differences in human behavior. In contrast, most existing data-driven approaches require data labeled by experts. In both cases, systems quickly become highly complex. For example, Okoshi et al. [58] proposed a system with multiple hierarchies of classifiers that detect breakpoints in activities based on previously detected postures or movements [5, 58]. Finally, existing AMSs manage interruptions quite simply, i.e., by delivering them whenever task boundaries are detected or choosing between multiple interruption modalities (such as visual, auditory, or haptic) [5].

### 2.3 Computational Rationality

A theoretical concept that may be able to more precisely model human multitasking behavior while accounting for various uncertainties and individual differences is computational rationality [59]. In contrast to classical RL (where agents are typically trained to achieve superhuman performance, for example, to play computer games [53]), CR introduces constraints that incorporate human limitations to make agents behave similarly to humans. To provide an example related to this paper, while a classical RL agent would always see the true environment and react instantly to a task switch, a CR agent can incorporate constraints related to human visual perception to provoke a reaction time. This can be achieved by distinguishing between an internal (i.e., the agents’ “mind”) and an external environment in which it operates. The CR agent (i.e., the modeled user) interacts with the internal environment via observations and actions and with the external environment via stimuli and responses. The transition from the internal to the external environment involves the mentioned constraints. To build upon the example above, an external stimulus (such as a task switch making the agent focus on another platform) could lead to inaccurate observations due to limitations in human vision. The subsequent action is performed based on the inaccurate internal state formed by the observation. The response to the external environment can again be subject to constraints (for example, limited by delays that mimic the human locomotor system). The internal environment can include psychological constructs such as memory (and their intrinsic limitations), signal noise, uncertainty, emotion, or stress and is updated based on the partially observable external environment. The internal representation can be modeled as a probability

distribution over the possible states. With the help of a partially observable Markov decision process (POMDP) [8] (see 3.2) it can be described that the environment is only partially observable and the agent acts based on a belief state rather than the real state of the environment. The theory is called “Computational Rationality” because the control policy used for the action selection is the “rational” (or optimal) policy taking into account the limits of the cognition, and the theory assumes that real humans would indeed perform optimally given their limitations. The great advantage of CR is its ability to simulate not only general but also individual behavior. Since the included constraints can be parametrized, the user model can be configured to perform stronger or weaker. Both an appropriate definition and parametrization of these constraints are necessary to find a suitable model for individual humans. CR was already applied to different kind of tasks like visual search [19], multitasking [44], typing [42], pointing [7], menu selection [18], decision making [62] or drawing [73]. Further, Gebhardt et al. [29] used RL to model human task interleaving behavior, suggesting that RL could become “*a plausible model of supervisory control*”.

### 2.4 Summary

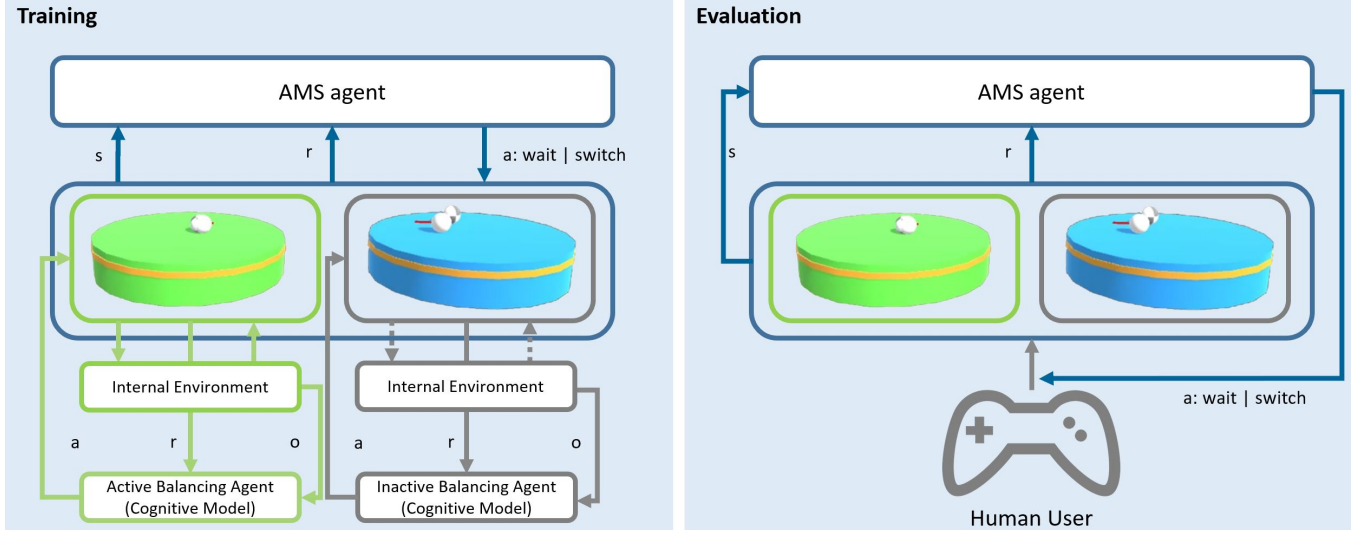
We have discussed the theoretical aspects of task switching and how existing AMSs have been designed so far. As argued, human multitasking is a highly complex process, where both theory- and data-driven AMSs quickly become highly complex and suffer various weaknesses such as inflexibility or difficulty in accounting for user differences. In contrast, computational rationality may be able to provide sophisticated models of human multitasking behavior that could be exploited by future AMSs.

## 3 TASK SWITCHING SUPPORT WITH REINFORCEMENT LEARNING

This work aims to evaluate the potential of an RL agent to develop an AMS that manages interruptions and controls human attention. We hypothesize, that an RL agent can learn a policy that automatically detects moments for interruptions and coordinates operator resources (i.e., automatically and properly assigns tasks) in a way to improve overall performance.

### 3.1 The Balancing Game as Example of an AMS-Supported Dual-Task

Existing experiments on multitasking have used a multitude of natural (i.e., texting/chatting, data entry, driving, reading comprehension, object identification, etc.) and standardized tasks (tracking,



**Figure 3: The architecture consists of the RL AMS agent that has unlimited access to the (virtual) environment and decides when to operate on one of the two platforms. To simulate a user playing the game, we implemented balancing agents with cognitive constraints for training the algorithm (left). In the subsequent evaluation phase (right), the AMS agent used the learned policy to switch between platforms when humans were playing the game.**

n-back, VCR task, pursuit, etc.) [13, 17, 55] with varying perceptual, cognitive, and physical demands that require skill-, rule-, and knowledge-based performance [65]. To evaluate our concept we have set up a novel continuous, motoric, fast-paced dual-task. Users control a computer game where they have to balance two balls on two dedicated platforms using the left stick on a gamepad (see Figures 1 and 3). Only one platform can be controlled at each moment in time to simulate a situation where users have to switch between the two platforms (i.e., tasks) sequentially.

Considering the multitasking continuum, this task shows aspects of both, sequential and concurrent multitasking (i.e., players can see both platforms all the time but can only operate on one platform, see Figure 2). We argue that the continuous real-time nature of the task involves visual-cognitive demands, requires both skill-based and strategic components, and can act as a proxy for similar tasks (for example, the lane-keeping task in driving research [57]). Further, performance in the task can easily be assessed concerning the game score (i.e., the duration the balls are successfully balanced on the platforms), which allows the formulation of a simple reward. Finally, in its monotask version, the balancing task has a comparably small state space and allows for fast training.

In our dual-task version of the game, users can either switch between the platforms themselves (using the shoulder buttons of the gamepad) or are assisted by the developed AMS. The difficulty level steadily increases by reducing the drag value of the balls over time. This value defines the decay rate of the ball’s linear velocity. Therefore this value is used to simulate drag, air resistance, or friction. The starting value is 0.8 and is reduced every 5 seconds by a value of 0.025, where negative drag values are allowed so that the ball would get faster over time. If the ball moves slowly, a force towards its current direction is applied to prevent them from lying still which would mean that no further adjustment of

the rotation of the platforms would be necessary. When a ball falls off the platform, the game ends. This design allows a clear measurement of performance (i.e., the longer the game time, the higher the score), which is relevant for training the RL agents and quantifying users’ performance in the subsequent evaluation. We implemented the game in Unity and utilized the ml-agents library [45]. Every episode starts with a 5 seconds lasting countdown. The balls fall down from a random position inside the inner third of the platforms. An episode ends in case one of the balls falls off the corresponding platform. When assisted by the AMS, a switch is announced by a short audio signal as well as by coloring the target platform white. This announcement is performed 0.3 seconds before the actual switch. After a switch, the source platform moves towards its horizontal rotation with a speed of 10 degrees per second.

We designed an algorithm that supports users by automatically switching between the platforms at appropriate moments. Strictly speaking a “moment” is nothing else than a particular configuration of the environment, with time being a series of subsequent environment states. Configurations that are suitable for a task switch, in turn, can be considered as moments. Thereby, the AMS agent is supplied with a comprehensive representation of the problem space, including the environments of the involved tasks (see Figure 3). Based on this state space, the agent acts by either waiting (letting the user continue the task in the foreground) or switching to a particular task that demands attention. The agent is rewarded by interrupting and assigning the user properly (i.e., increasing the game score, see Section 3.3 for implementation details).

To be able to train the AMS agent, we designed a computationally rational balancing agent with cognitive constraints using RL to simulate a human playing the game (balancing agents, see Figure 3, left). The CR agent does not observe the true but an internal environment [35], where various aspects (i.e., ball positions and



speeds) are included in a belief state. The AMS agent on top activates either the left or right balancing agent via task switches. For example, when a balancing agent is deactivated by the AMS agent (i.e., switching to the other platform), the balancing agent is cut off from the real environment and gradually “forgets” the true position/velocity of the ball, leading to a deviation of the true state and the belief state. This is illustrated on the right (blue) platform in Figure 3. Here, the white ball represents the believed position and velocity of the ball, which is different from the true ball in the background. When the AMS agent switches to a platform, the corresponding balancing agent is activated, and the belief state is gradually updated again, simulating a regain of attention and reaction time of a human. After some (reaction) time the true and believed ball position match closely again (Figure 3, left). By using the constraint balancing agents, the AMS agent is forced to train a task switching policy adjusted for humans. The CR balancing agent is adapted based on different parameters to describe to what extend a certain constraint is present (for more details see 3.5). Consequently, diverse parameter assignments represent CR models that engage in the balancing game tailored for different individuals.

After training, the policy learned by the AMS agent can be used for automated task switching when real humans play the game. This means, that the CR model for the balancing agents are cut off from the system as the balancing task is solely performed by the human (see Figure 3, right). Additional technical details about the ml-agents training process in combination with Unity can be found in the supplementary materials.

### 3.2 Markov Decision Process

In RL, an agent performs an action  $a$  in a state  $s$  of an environment at time step  $t$ , which results in a change to the successor state  $s'$  with a probability of  $p$  to reach a certain predefined goal by maximizing a reward signal  $r$ . This work includes two different agents which can be distinguished by the observability of the state space. The AMS agent can access the true state space describing the current conditions of the platforms. This completely observable RL problem can be defined based on a Markov decision process  $MDP(S, A, P, R, \gamma;)$  where the upper case letters describe the sets of the former mentioned corresponding elements.  $\gamma$  describes a discount factor where  $0 \leq \gamma \leq 1$  and is used to weight recent rewards more strongly than rewards from the past. Usually, RL algorithms estimate so-called *value functions* which indicate the expected return an agent gets in a certain state. Value functions are defined based on the behavior of an agent, called *policy*  $\pi$ . The goal of RL is to learn a policy  $\pi$  to maximize the value function and, therefore, the expected reward. The policy is defined as a mapping of states  $s$  to the probabilities of the selection of certain actions  $a$  in this state, such that  $\sum_{a \in A} \pi(a|s) = 1$ . Finally, based on this information, the *state-value function* can be defined [77]:

$$v_{\pi}(s) = \sum_{a \in A} \pi(a|s) \sum_{s' \in S} \sum_{r \in R} p(s', r|s, a) [r + \gamma v_{\pi}(s')] \text{ for all } s \in S \quad (1)$$

Therefore, to calculate  $v_{\pi}(s)$ , the rewards  $r$  of the different possible successor states  $s'$  must be summed with their corresponding expected rewards  $v(s')$  resulting in a recursive calculation. The

values of  $v(s)$  for all  $s \in S$  can be learned in an iterative way and saved in a table starting with initial values of 0. This work has a continuous state space that prevents tabular methods from working. Therefore, it uses approximation techniques like neural nets to generalize this continuous state space.

The former definition assumes that the agent is aware of its current state but there are scenarios where this assumption does not hold due to the only partial observability of the environment. This is the case for the cognitive balancing agent which can only observe its state if the platform is currently active otherwise the uncertainty over the state space grows over time leading to imprecise actions. Those problems are defined as a partially observable Markov decision process  $POMDP(S, A, P, R, \gamma; H, \Omega, O)$  where  $S, A, P$  and  $R$  are equal to the definition of the  $MDP$ .  $\Omega$  is a finite set of observations  $o \in \Omega$  the agent can perceive.  $O$  denotes the *observation function*.  $O(s', o, a)$  describes the probability that an agent gets in state  $s'$  making observation  $o$  after performing action  $a$ . Since the agent does not know the true state, it must aggregate a belief state  $b(s)$  based on the previous experience. The belief state is updated via the following equation:

$$b'(s') \propto O(s', o|s, a) \sum_{s \in S} p(s'|a, s) b(s) \quad (2)$$

Here,  $p(s'|a, s)$  denotes the transition probability to state  $s'$  after performing action  $a$  in state  $s$ . In the context of this work,  $O(s', o, a)$  is calculated based on a fixed parameter  $o_p$  like in Formula 5. A belief state  $b$  is a probability distribution over  $S$ , such that  $\sum_{s \in S} b(s) = 1$  and  $0 \leq b(s) \leq 1$ . This probability distribution is updated during the execution of the environment. The policy uses  $b(s)$  instead of  $s$  since  $s$  is not known to the agent.

### 3.3 AMS Agent

This agent perceives the environments of the involved tasks (as states) and can either stay on the current platform or switch to the other (via actions). By training, it determines a policy so that switches occur in a way that maximizes the game score (reward, see Figure 3). The problem environment of the task switching challenge can be defined as a Markov decision process  $MDP(S, A, P, R)$ . The state  $s \in S$  of the environment is defined by the coordinates of the balls, the angles of the surfaces, and the movement direction, or velocity, of the ball. The action  $a \in A$  is defined by either switching to a new platform or staying at the current one. This decision is requested every 200 ms.  $p(s'|s, a) \in P$  describes the probability that the player moves the surface in a certain direction resulting in the successor state  $s'$ . Therefore, only the sub-environments of the problem are stochastic. However, the selection of the instance by action  $a$  directly causes the switch to the respective instance. Therefore, this selection process is deterministic, with  $p(s'|s, a) = 1$ . The reward  $r \in R$  is defined by the following function:

$$r_t = \frac{1}{1 + e^{-\Delta t_s - 0.5}} \quad (3)$$

where  $\Delta t_s$  describes the elapsed time since the last switch. Based on this function, fast switches result in a smaller value than those that are performed after one second leading to less often performed switches. This adjustment was made to avoid unnecessary switching observed in situations where both balls were near the center of

the platform (i.e., where switching would not result in any change in the reward signal). Consequently, we utilized a logistic function so that the agent avoids unnecessary switches within one second.

### 3.4 Balancing Agents

Through the application of the principle of computational rationality [35], we were able to train the AMS without the need for human involvement. Therefore, we built and refined two distinct models: an *unconstrained* one and a model incorporating human cognition (*cognitive* model).

**3.4.1 Unconstrained Model.** This model defines a ball balancing agent without any human constraints that can, therefore, achieve superhuman performance. This agent is used as a baseline to examine if any of the human cognition adaptations affect its capabilities to find the opportune moment for a task switch when used as input for the training of the AMS agent. It uses a sensor vector with a space size of 8 and the same observations as the AMS agent except that it only collects the information from its own platform. The action vector is continuous, has a size of 2, and defines how much the platform should be rotated on the x- and z-axis per action. It ignores the y-axis. The reward signal  $r$  at time  $t$  is defined by the following formula:

$$r_t = 1 - \frac{D_t}{radius} \quad (4)$$

Here  $D_t$  is the distance of the ball to the center of the platform at time  $t$  and  $radius$  is the radius of the platform.

**3.4.2 Cognitive Model.** The human modeling agent mimics the behavior of a human playing the balancing game. Human players are subject to certain constraints in perceiving the actual state of the game. The following constraints were defined:

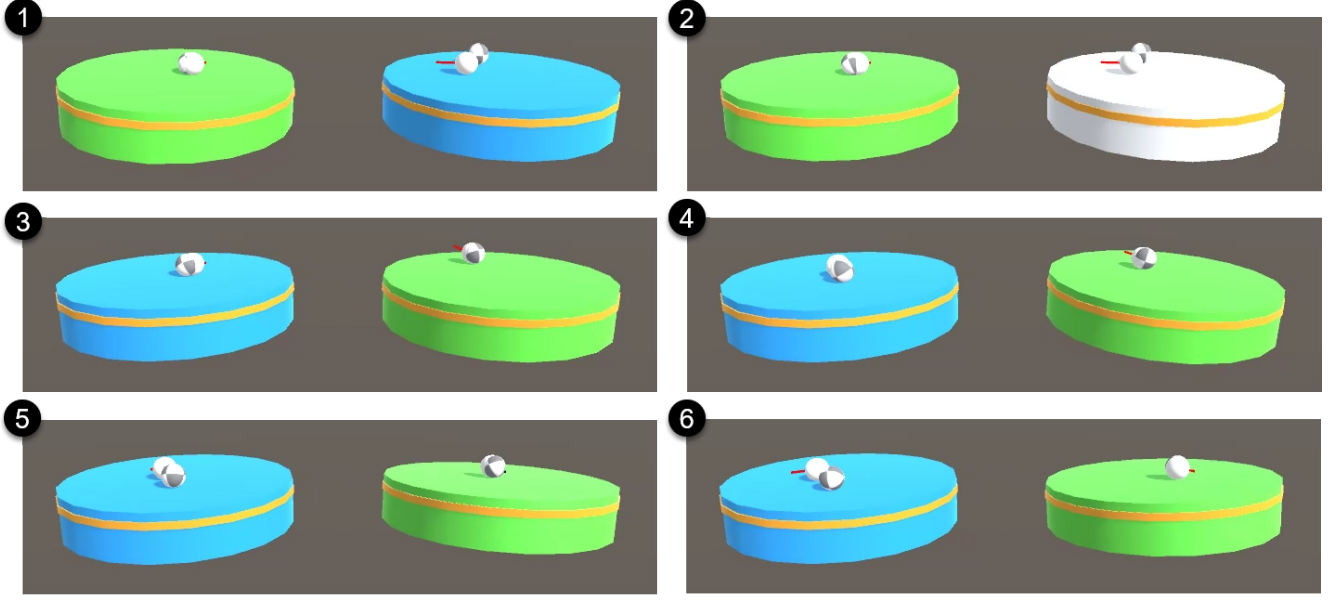
1. Adaptation of the internal representation (velocity and position) of the ball after a task switch.
2. Imprecise values of the following objects due to limitations in visual perception:
  - 2.1. Position of the ball
  - 2.2. Velocity of the ball
  - 2.3. Angle of the platform

The imprecise position, for instance, can be explained by the perception being more challenging due to the camera position (camera angle to the platform). The adaption of Point 1. depends on the difference between the states of the platforms, s.t. the modification of the internal representation towards the true state is fast if both states are similar (similar ball velocity and position per platform) or slow if they are significantly different. Furthermore, not only their similarities are relevant. It could be the case that a human remembers the approximate state since the last task switch. Therefore, the uncertainty of the state of the currently inactive platform grows over time. This noise could cause a non-ideal or delayed action by humans. These restrictions and the underlying internal representation can be defined by a partially observable Markov decision process  $POMDP(S, A, P, R, \Omega, O)$ . The state space of the balancing agent is continuous, s.t. the belief state  $b(s)$  could be either described with a density function or based on a discretization into bins. To reduce the computational complexity, this work

uses a discretization of the ball position and velocity to model the belief state  $b(s)$ . The transaction probabilities  $p$  in Equation 2 are calculated based on a normal distribution of the ball velocity with a fixed parameter for the standard deviation  $\sigma_p$  and a mean value of a random value of an additional normal distribution. This random value is determined based on a fixed parameter for the standard deviation  $\sigma_{mean}$  and the mean value of the estimated velocity. The estimated velocity equals the true velocity if the platform is active. Otherwise, in the first step, the estimated velocity equals the random value of the additional normal distribution. The second step uses a new random value from the additional normal distribution with the received random value from the first step as its mean value. Therefore, the estimated velocity gets more imprecise with every further update step due to the use of a random value from the additional normal distribution as the mean for the original distribution. Accordingly, the determination of a random value as the mean for the original distribution is used to generate a further deviation from the last observed velocity.  $\sigma_{mean}$  is reduced every update step by a division of 2 according to the fixed update cycle of Unity. Consequently, the deviation from the actual velocity is determined shortly after the task switch and the trajectory of the belief state remains constant after an initial uncertainty. The observation probability  $O$  in Equation 2 is calculated based on a fixed observation probability parameter  $o_p$ :

$$O(s', o|s, a) = \begin{cases} o_p & \text{if } s' \text{ is the true new state and platform active} \\ \frac{1-o_p}{\#s-1} & \text{if } s' \text{ is not the new state and platform active} \\ 1 & \text{if the platform is not active} \end{cases} \quad (5)$$

Here,  $0 \leq o_p \leq 1$  applies. This definition of the observation probability leads to a fast update of the belief state to the true state if the platform is active and  $o_p$  is not small. If the platform is not active, no observation is made, and therefore,  $O(s', o|s, a) = 1$  for all  $s' \in S$ . Figure 4 illustrates the change of the belief ball position over time. The update of the belief state (see formula 2) is performed during the execution of the environment in a predefined interval  $u$ . This is a new approach since older work calculated the transitions of the probability distributions in advance [42]. The update of the probability distribution is calculated in parallel with the help of the *Unity C# Job System* in combination with the *Burst Compiler* [23]. The platform is discretized into 1000 bins. The new probability that the ball is inside one of these bins can be calculated independently by taking samples describing possible velocities from the normal distribution. A visitor bin  $j$  can be determined by subtracting this sample velocity from the current position of a certain bin  $i$  s.t.  $position_j \leftarrow position_i - velocity$ . Iteratively all possible visitors are calculated and their current probabilities are used to determine the updated probability of bin  $i$  by  $New[i] \leftarrow New[i] + \frac{Current[j]}{n_s}$  where  $New$  describes a vector with the updated probabilities,  $Current$  the current probabilities and  $n_s$  the number of samples. The resulting probability that the ball is inside bin  $i$  is multiplied by the observation probability described in formula 5. Furthermore, the distribution of the previously active platform is considered for a certain time after a switch (described by  $d_t$ ), to model the slow adaption of humans to a fast-changing environment. Therefore, the agent performs actions based on the



**Figure 4: The change of the belief ball position over time. The white ball shows the currently believed ball position. The red line indicates the velocity with the highest probability of the normal distribution. The black line points to the average direction of the sample velocities. At step 1, the left platform is active (indicated by the green color). At step 2, an upcoming task switch is indicated by the white color of the target platform. At step 3, the task switch is performed resulting in an update of the belief ball position of the right platform: from an imprecise position in step 2 to the true position at step 3. In steps 3 to 6, the increasing mismatch, between the belief ball position and the true ball position on the left, inactive platform, can be seen. This ever-increasing mismatch is caused by the determined sigma values of the normal distribution.**

belief state of the previously active platform on the currently active platform for  $d_t$  seconds after a switch. This approach is based on the assumption that a human player remembers the state of the previously active platform and its increasing inaccuracy over time but it does not model the chance that a human performs an action that was intended for the old platform after a fast task switch. This can be modeled by the introduction of a further parameter  $t_{od}$  describing the duration an agent performs an action based on the belief state of the former active platform.

### 3.5 Training and Parameter Inference

The AMS agent was trained based on a cognitive model defined by its parameters. They were determined by an inverse modeling approach, which describes the process of inferring parameters from behavioral data [56]. Therefore, the behavioral data of a single participant with average gaming skills (value of 4 on a scale from 1 to 7, where 1 is "I Don't play at all" and 7 is "Professional Level") was collected by playing the game for 500000 actions with a "random" AMS (i.e., random switching interval with a minimum of 0.2s) and a constant difficulty level (fixed ball drag value of 0.8). 2000 different parameter assignments were evaluated and compared to the human behavior via a distance function. The parameterization of the model most similar to the human behavior was used for training and evaluation of the AMSs (Standard Deviation  $\sigma_p = .01$ ; Standard Deviation  $\sigma_{mean} = .2$ ; Update Period  $u = .4$ ; Observation Probability  $o_p = .4$ ; Old Distribution Persistence Time  $d_t = .1$ ). The

balancing agent was trained for 10 million steps in combination with the AMS agent which was trained for 5 million steps. A step is defined by a single decision. The training of both agents converges after the corresponding steps. More technical details about parameter inference and the training process can be found in the supplementary materials.

## 4 USER STUDY

### 4.1 Experiment Design

We evaluated the AMS prototypes in a lab experiment. Our main hypothesis addresses the question of whether participants supported by the AMS achieve higher performance than when they decide which tasks to attend at each moment. Further, we aimed to investigate if the AMS would work better when being trained on the user model with cognitive constraints, compared to the unconstrained model. Finally, we wanted to know how the AMS would perform when task switches are not automatically performed but indicated to the user via notifications. This yielded the following four conditions, which were tested in a quasi-randomized within-subjects experimental design:

- **Cognitive Model:** In this condition, participants played the game with the AMS trained on the user model with cognitive constraints. The AMS automatically switched between the platforms based on the learned policy (see Section 3.4.2).



- **Unconstrained:** In this condition, participants played the game with the AMS trained on the unconstrained user model. Again, the AMS switched between the platforms based on the learned policy automatically (see Section 3.4.1).
- **Notification:** We wanted to know if the AMS would also benefit users when only notifying them. In this condition, the AMS used the policy trained using the *cognitive* model. However, task switches were not enforced automatically but had to be confirmed by the user by pressing a button. To allow for suitable comparisons, we decided to automatically switch to the target platform in case the user does not confirm within one second.
- **No Supervisor:** In this condition, no AMS was present and the task of switching between the platforms was solely performed by the participants. This condition acts as the baseline control condition for investigating the main hypothesis.

Strictly speaking, the evaluation procedure did not include the balancing models used for training the AMSs. For the sake of simplicity, we still call the corresponding conditions based on the names of the underlying user models, i.e., condition *cognitive model* refers to the AMS trained on the cognitive model, while the condition *unconstrained* refers to the AMS trained using the unconstrained model.

## 4.2 Participants

In total,  $N=43$  participants (30 male, 12 female, one preferred not to say,  $M = 27.73$ ,  $SD = 4.21$  years; mainly students and university staff) completed the experiment. Participants self-rated computer/video game skills were close to the theoretical mean on the Likert scale from 1 to 7 ( $M = 3.83$ ,  $SD = 1.88$ ,  $Median = 4$ ,  $IQR = 3$ ). Regarding the amount of playing hours, the median answer was <1 hour per week. Participants' tendency to actively engage in or avoid intensive technology interaction measured by Affinity for Technology Interaction (ATI) scale [26] showed above neutral result ( $M = 5.25$ ,  $SD = 0.98$ ).

## 4.3 Procedure

The experiment was performed directly in the Unity Editor. Participants used a DualShock 5 gamepad, specifically its joystick, to control the angles of platforms and the shoulder buttons to switch between the platforms in the *notification* or *no supervisor* conditions. The research was carried out within a laboratory setting, where we provided guidance to the participants throughout the study. Upon arrival, participants were briefed about the study and were asked for demographic data. Next, they completed a short training in each condition in quasi-randomized order (to account for order effects, one minute per condition). This was followed by the four experimental conditions, again in quasi-randomized order (following an experiment plan where 4 conditions result in  $4!=24$  different arrangements, i.e., every 24 participant, all combinations of conditions appeared exactly once, and participants were assigned in the order they appeared). In each condition, we collected the results of five episodes that lasted longer than 5 seconds (i.e., we removed shorter episodes since sometimes the random ball positions at the start of the episode were disadvantaging the player; the 5s were determined experimentally when configuring the difficulty

of the task before running the experiment). Participants were told that the experiment would end automatically once enough data was collected. After each condition, participants completed a short post-condition survey. At the end of the study, the data collection was supplemented with a post-experiment survey and a short semi-structured interview. Overall, the experiment lasted for about one hour per participant.

## 4.4 Measurements

To evaluate the discussed AMS and the baseline condition, we collected performance- and behavior-related data as well as subjective ratings as follows:

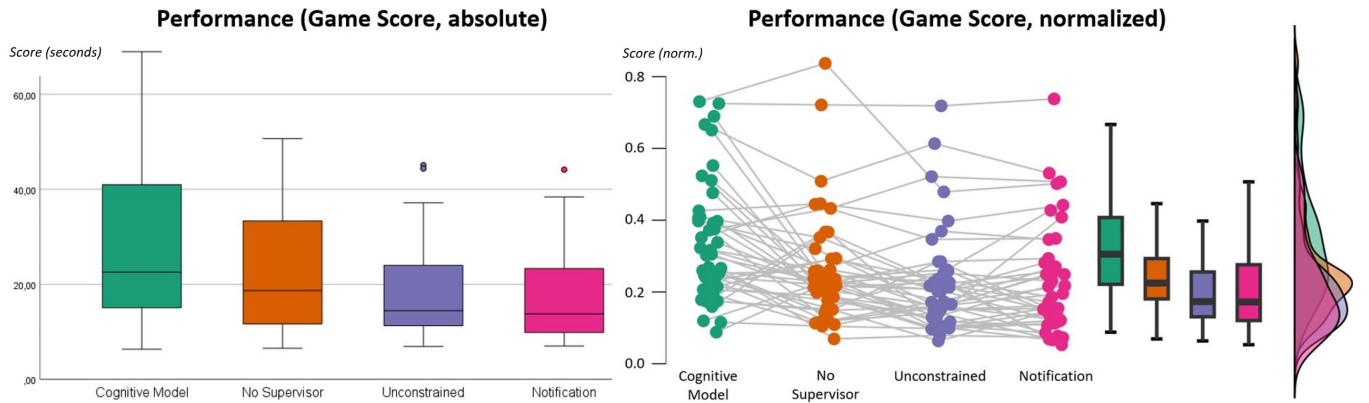
- **Performance** was assessed using participants' final game score (i.e., time until the game was over) after each episode. Since the difficulty of the game steadily increases, the score represents a valid measure of participants' capability to balance the ball on both platforms. Another indicator of performance is the ball's average distance to the platform center during an episode. Assuming that a good player would balance the ball close to the platform center, low values indicate high-performance and high values indicate low one. We calculated the geometric means for players' repeated performance in the respective conditions to be more robust against potential outliers [25, 71].
- **Task Switching Behavior** was assessed with multiple parameters. We collected the average time spent on a platform between the task switches (average time on task) and participants' reaction time. The reaction time was defined as the time between a task switch and the subsequent first joystick action, i.e., the threshold of the Euclidean distance between a zero vector and the averaged performed joystick input (a threshold of 0.19, similar to reaction time measurements in automated vehicles [20]). Further, we specified a set of values describing the state of the balancing game at the moment of a task switch. The source platform rotation describes the state of the currently active platform before the task switch (in degrees,  $\text{deg}^\circ$ ). The ball distances (in Unity units, u) and velocities (in Unity units per second, u/s) are other indicators of the task-switching behavior. Lower angle, distance, and velocity values on the source platform would mean that the situation was "cleared" before a task switch, while higher distance and velocity values at the target platform indicate a need for more urgent reactions.
- **Subjective Scales** were included to assess participants' workload and overall experience. After each condition, the workload was measured with the short version of the NASA-TLX questionnaire [31]. Further, participants rated their subjective experience after each condition on a single 7-point Likert scale item ("Please rate your overall experience of playing under this condition, from *don't like at all* to *like it very much*").

## 5 RESULTS

To evaluate the results between the four conditions, we used IBM SPSS V.24 to conduct Friedman tests with subsequent pairwise Bonferroni-corrected Wilcoxon tests or repeated measures ANOVA

**Table 1: Players’ performance and subjective ratings in the different conditions. The absolute score resembles the game time in seconds until the game is over, the normalized score is based on players’ maximum performance over all episodes, and the distance to the center shows the average deviation of the balls to the platform centers in Unity units (i.e., “meters”).**

	Cognitive Model M (SD)	No Supervisor M (SD)	Unconstrained M (SD)	Notification M (SD)
<b>Performance Data</b>				
<b>Absolute score (s)</b>	36.29 (35.17)	25.99 (27.05)	19.71 (16.10)	20.03 (18.47)
<b>Normalized score</b>	.34 (.17)	.26 (.15)	.22 (.14)	.23 (.15)
<b>Distance to center</b>	3.12 (.30)	3.19 (.25)	3.11 (.26)	3.21 (.25)
<b>Reaction time (ms)</b>	346 (151)	323 (179)	306 (94)	314 (143)
<b>Subjective Workload</b>				
<b>Mental demand</b>	3.68 (1.57)	5.02 (1.39)	4.07 (1.53)	4.91 (1.44)
<b>Physical demand</b>	2.32 (1.29)	3 (1.75)	2.41 (1.50)	2.84 (1.80)
<b>Temporal demand</b>	4.02 (1.34)	4.09 (1.70)	4.55 (1.55)	5.18 (1.59)
<b>Effort</b>	4.11 (1.50)	4.96 (1.31)	4.34 (1.52)	5.11 (1.45)
<b>Frustration</b>	3.86 (1.67)	3.59 (1.62)	4.66 (1.31)	5 (1.46)
<b>Subj. performance</b>	4.25 (1.28)	3.93 (1.56)	3.71 (1.05)	3.14 (1.44)
<b>User Preference</b>				
<b>Overall rating</b>	4.39 (1.48)	4.43 (1.56)	3.48 (1.23)	3.05 (1.54)



**Figure 5: Participants’ absolute (left) and normalized (right) game scores in the four experimental conditions. A higher score indicates better performance. The normalized scores (based on players’ maximum performance over all episodes) allow for a better visual interpretation of the data since the absolute scores strongly varied between the participants.**

with post-hoc tests (depending on the data being normally distributed or satisfying the central limit theorem). Additionally, we utilized JASP [41] to generate Raincloud diagrams for our data and we confirmed our results with Bayesian repeated measures ANOVA. Finally, we conducted some exploratory comparisons between our highest-performing AMS agent (based on the *cognitive* model) and participants’ self-chosen task switching strategy (*no supervisor*).

## 5.1 Performance

Descriptive statistics and data plots of participants’ absolute and normalized game scores in the individual conditions can be seen in Table 1 and Figure 5. In contrast to the monotask version of the game (where 12 players could balance the ball on a single platform and without task switches for  $M = 103.24$   $SD = 56.25$  seconds, see supplementary materials for details), participants of this study

performed, depending on the conditions, three to four times worse in the dual-task versions.

Regarding participants’ performance when playing in the four different conditions (the *unconstrained* and the *cognitive* models as well as the *notification* and *no supervisor* conditions), an ANOVA indicated a significant effect (Greenhouse-Geisser;  $F(1.69, 70.82) = 11.534$ ,  $p < .001$ ,  $\eta_p^2 = .215$ ). Pairwise comparisons with Bonferroni correction reveal that participants performed significantly better with the *cognitive* model compared to the *unconstrained* ( $p < .001$ ), the *notification* ( $p = .004$ ), and the *no supervisor* ( $p = .042$ ) conditions. In contrast, in the *no supervisor* mode players performed better than the AMS agent trained with the *unconstrained* model ( $p = .043$ ). This result was confirmed with a Bayesian ANOVA

( $BF_{incl} = 15579.798$ ), with the *cognitive* model showing higher performance than the *unconstrained* model ( $BF_{10,U} = 179.19$ ), the *notification* mode ( $BF_{10,U} = 40.18$ ), and the *no supervisor* ( $BF_{10,U} = 5.36$ ) mode.

Another potential indicator of performance is the distance of the balls to the platform centers (averaged over an episode). In other words, high performance would mean that the balls are quite close to the platform centers, while lower performance would be indicated by larger distances. Although the ANOVA suggests a significant effect (Assumption for sphericity met;  $F(3, 126) = 3.974$ ,  $p = .01$ ,  $\eta_p^2 = .086$ ), none of the Bonferroni-corrected pairwise comparisons was significant (see Table 1 for the quite similar means in the respective conditions).

Further, we investigated how participants' individual performance influenced the results. While 32 participants showed the highest performance when playing with the AMS agent trained on the *cognitive* model, 11 showed their best results in the *no supervisor* condition. 8 of these players reside in the upper half when performing a median split on their performance (in the *no supervisor* condition), compared to 3 in the lower half – in other words, the AMS worked better for low-performing than for highly-skilled players. Finally, we evaluated the order of conditions, in particular, whether a learning effect occurred that made participants perform better over time. However, a repeated measures ANOVA indicated no significant effect, and directly comparing the first ( $M = 23.4$ ,  $SD = 29.0$ ) and the last try ( $M = 23.0$ ,  $SD = 18.1$ ) independent of the experimental condition yielded no significant difference.

## 5.2 Task Switching Behavior

Regarding participants' reaction times after switching the platform, we did not find any statistically significant differences between the experimental conditions (see Table 1). Further, we hypothesized that the reaction time could depend on how urgent a reaction is needed. For that, we calculated the set of task switching behavior measures as described in section 4.4. We hypothesized that the reaction times are partly depending on the difficulty of the upcoming task, i.e., a fast-moving ball with a high distance to the center is more difficult to resolve and needs a faster reaction than a slow ball placed in the center of the target. For both the balls' velocity and distance to the center, we observed a significant but weak negative correlation with reaction times (velocity:  $R = -.16$ ,  $p < .001$ ; distance:  $R = -.083$ ,  $p < .001$ ).

Additionally, we investigated if the task switching behavior of our best-performing AMS agent (the one trained using the *cognitive model*) differs from the participants' self-chosen interruption strategy (*no supervisor* condition) using Wilcoxon signed rank tests, see Figure 6. Participants stayed significantly longer on a platform when playing without an AMS ( $M = 4.46s$ ,  $SD = 1.61s$ ) compared to the *cognitive model* ( $M = 2.66s$ ,  $SD = .52s$ ;  $Z = -5.458$ ,  $p < .001$ ,  $BF_{10} = 9654.518$ ). The rotation of the source platform (*cognitive model*:  $M = 9.36^\circ$ ,  $SD = 4.47^\circ$ ; *no supervisor*:  $M = 10.04^\circ$ ,  $SD = 4.02^\circ$ ;  $BF_{10} = .292$ ) and the ball distance at the target platform (*cognitive model*:  $M = 2.95u$ ,  $SD = .25u$ ; *no supervisor*:  $M = 2.97u$ ,  $SD = .39u$ ;  $BF_{10} = .177$ ) did not differ.

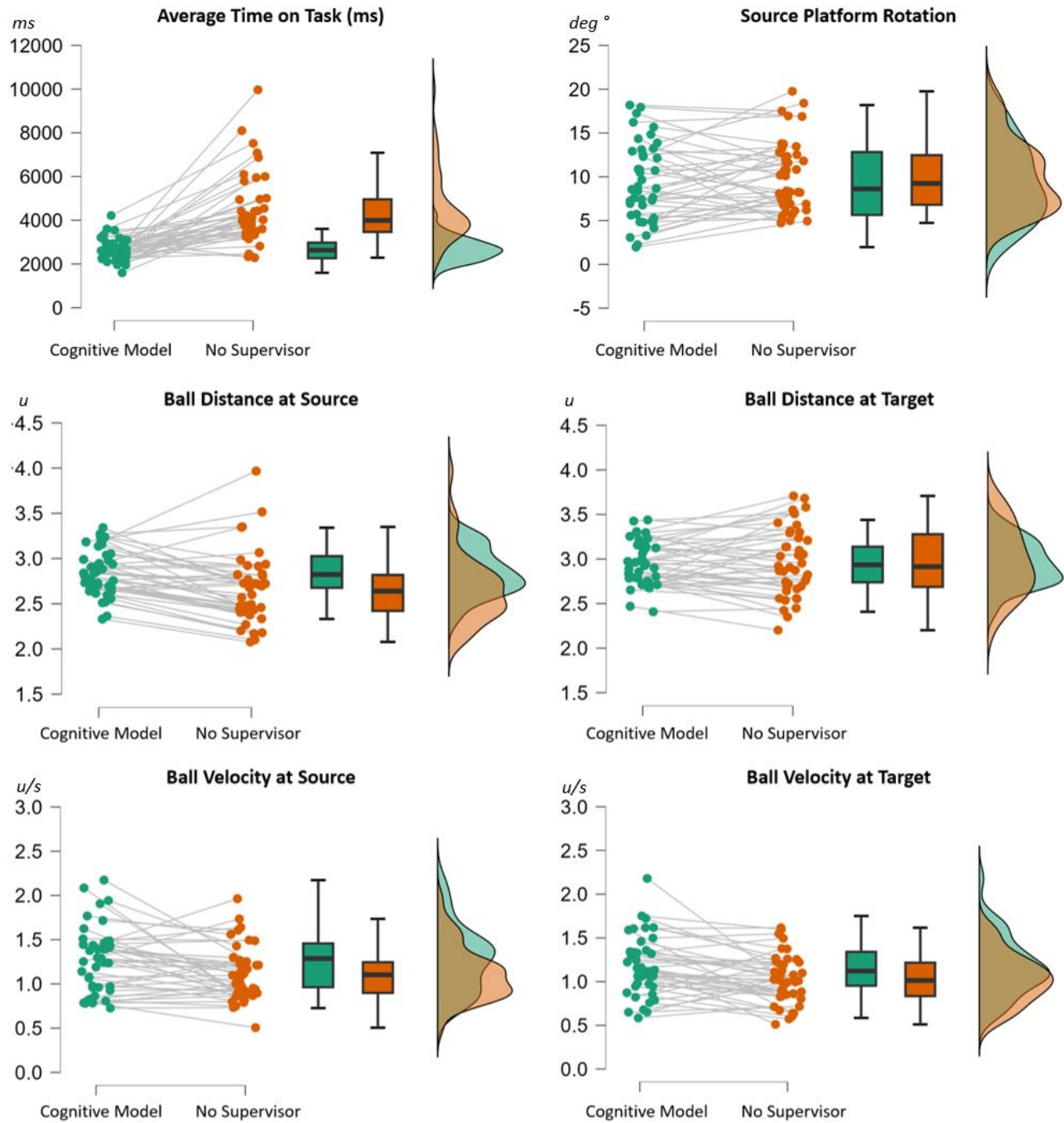
The *cognitive model* ( $M = 2.85u$ ,  $SD = .25u$ ) switched at significantly larger deviations of the ball on the source platform than the participants themselves (*no supervisor*:  $M = 2.66u$ ,  $SD = .39u$ ;  $Z = -3.770$ ,  $p < .001$ ,  $BF_{10} = 173.515$ ). Further, participants tended to slow down the ball on the source platform to a significantly lower speed ( $M = 1.12u/s$ ,  $SD = .30u/s$ ) before switching than the *cognitive model* (velocity at source,  $M = 1.28u/s$ ,  $SD = .37u/s$ ;  $Z = -2.570$ ,  $p = .01$ ,  $BF_{10} = 12.902$ ), while the *cognitive model* tended to switch at significantly higher ball speeds on the target platform ( $M = 1.17u/s$ ,  $SD = .34u/s$ ) than the participants in the *no supervisor* condition ( $M = 1.03u/s$ ,  $SD = .28u/s$ ;  $Z = -2.795$ ,  $p = .005$ ,  $BF_{10} = 14.912$ ). Overall, the results suggest that participants switched more cautiously between the platforms themselves (*no supervisor* condition) than when using the best-performing AMS (*cognitive model*). They spent more time on the platforms and acted more carefully regarding a stable situation on the source platform before switching. Further, they switched at lower speeds of the ball on the target platform than the *cognitive model*.

Finally, we evaluated if the playing styles of less-skilled and higher-skilled players differ in the *no supervisor* condition using Mann-Whitney U tests (see Figure 7). Similar to the *cognitive model*, higher-skilled players switched more frequently and spent less time on a platform (*higher-skilled*:  $M = 3.87s$ ,  $SD = 1.15s$ ; *lower-skilled*:  $M = 5.06s$ ,  $SD = 1.80s$ ;  $U = 122$ ,  $p = .013$ ,  $BF_{10} = 2.875$ ). They also "cleared" the source platform closer to the neutral position before switching (*higher-skilled*:  $M = 8.33^\circ$ ,  $SD = .266^\circ$ ; *lower-skilled*:  $M = 11.75^\circ$ ,  $SD = .447^\circ$ ;  $U = 115$ ,  $p = .008$ ,  $BF_{10} = 2.557$ ). Also regarding the other parameters of the source platform, skilled players slowed down the ball more (*higher-skilled*:  $M = .94u/s$ ,  $SD = .18u/s$ ; *lower-skilled*:  $M = 1.29u/s$ ,  $SD = .30u/s$ ;  $U = 70$ ,  $p < .001$ ,  $BF_{10} = 113.238$ ) and kept it closer to the center before switching (*higher-skilled*:  $M = 2.46u$ ,  $SD = .23u$ ; *lower-skilled*:  $M = 2.87u$ ,  $SD = .41u$ ;  $U = 755$ ,  $p < .001$ ,  $BF_{10} = 34.361$ ).

## 5.3 Subjective Workload and Experience

Considering the subjective workload as assessed by NASA-TLX, Friedman tests indicated significant effects for all sub-dimensions (Mental Demand:  $\chi^2 = 33.85$ ,  $p < .001$ , Physical Demand  $\chi^2 = 14.01$ ,  $p < .01$ , Temporal Demand  $\chi^2 = 19.59$ ,  $p < .001$ , Effort  $\chi^2 = 20.99$ ,  $p < .001$ , Frustration  $\chi^2 = 25.02$ ,  $p < .001$ , Subjective Performance  $\chi^2 = 19.37$ ,  $p < .001$ ).

Pairwise Wilcoxon test (with Bonferroni-Holm correction) showed that playing with the AMSs trained on the *cognitive* ( $p < .001$ ) and the *unconstrained* ( $p < .01$ ) models has significantly lower mental demand, compared to the *notification* and the *no supervisor* conditions. Similar results were obtained for the physical demand. Also here, the *cognitive* and *unconstrained* models were rated significantly less physically demanding than *no supervisor* and *notification* ( $p < .05$ ). Regarding pace rush, both the *no supervisor* and the *cognitive model* conditions were rated significantly less temporally demanding than the *notification* condition ( $p < .001$ ). The *no supervisor* and *notification* conditions further received significantly higher scores for Effort than the two AMSs trained on the *unconstrained* and *cognitive* models ( $p < .05$ ). Participants felt significantly more frustrated with the *unconstrained* and the *notification* conditions than in the *cognitive* and the *no*

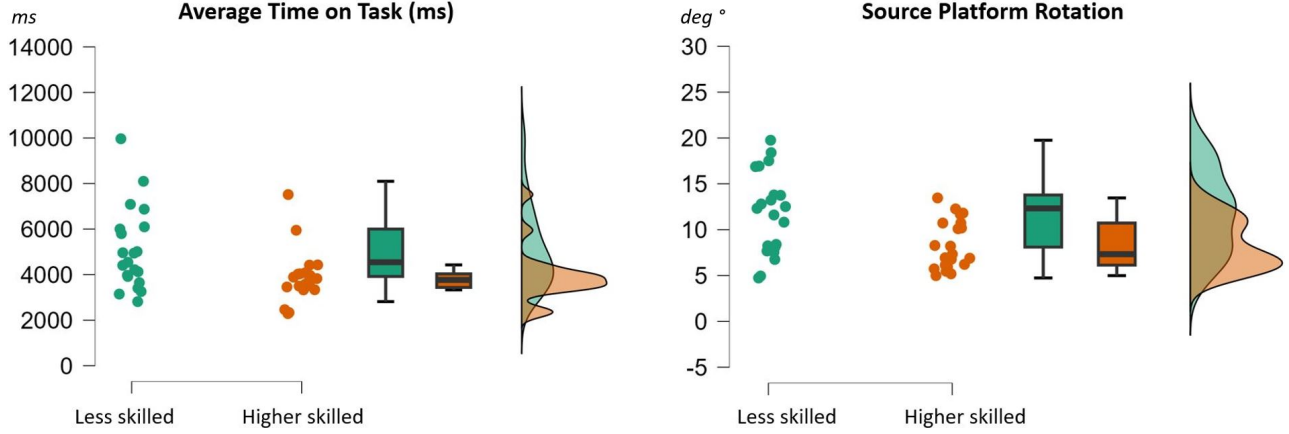


**Figure 6: Comparisons between the best-performing AMS trained using the *cognitive model* and the *no supervisor* condition. The results indicate that the AMS prompted participants to switch faster and resolve more difficult situations, while they acted more cautiously when playing without the system. Higher values for platform rotation (degrees), ball distance (Unity units), and ball velocity (Unity units/s.) indicate more difficult situations.**

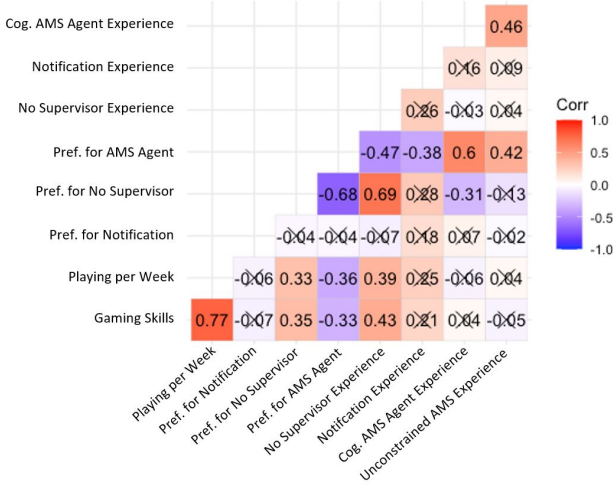
*supervisor* conditions ( $p < .01$  and  $p < .05$ ). Considering subjective performance, the *notification* condition had significantly lower ratings than the other conditions ( $p < .05$ ). Additionally, the *cognitive*

model had significantly higher ratings than the *unconstrained* one ( $p < .05$ ).

Finally, we assessed participants' overall preference with a single Likert-scale item ( $\chi^2 = 22.05, p < .001$ ). Thereby, both the



**Figure 7: Comparison of low and highly skilled players. Similar to the AMS, highly skilled players switched more frequently. They also “cleared” the source platform better before switching than the less skilled players. Higher values for platform rotation (in degrees) indicate more difficult situations.**



**Figure 8: Pearson Correlation Matrix between skills in games and mode preferences. Non-significant variables are crossed out.**

AMS trained with the *cognitive* model as well as the *no supervisor* condition received significantly higher scores than the other two conditions ( $p < .001$ ).

*Experienced Game Players prefer full under-human control mode.* The Pearson correlation matrix revealed a significantly higher preference towards the *no supervisor* mode by respondents with good computer/video game skills (they also tend to assess the subjective overall experience of *no supervisor* mode higher). This was accompanied by a significantly lower preference for the RL models (see Figure 8 for details).

*Multiple Linear Model of predicting overall experience explains almost 60% of its variability by physical demand, amount of hard work, level of stress or annoyance, and subjective success in accomplishing*

*the task.* Multiple linear regression was used to test if variables of NASA-TLX significantly predicted the overall experience regardless of the mode. For this, we combined the answers in all modes together. Akaike Information Criterion (AIC) was applied to find the best model fit. The overall regression was statistically significant ( $R^2$  adjusted = 0.584,  $F(6, 153) = 38.08$ ,  $p < .001$ ). We found that physical demand ( $\beta = -0.14$ ,  $p = .012$ ), effort ( $\beta = 0.17$ ,  $p = .006$ ), frustration level ( $\beta = -0.39$ ,  $p < .001$ ) and subjective performance ( $\beta = 0.58$ ,  $p < .001$ ) with intercept 2.85 ( $p < .001$ ) significantly predicted the overall experience. The final model explains 58.4% of overall experience variability.

## 6 DISCUSSION

The results of the experiment confirm our main hypothesis that **an RL-based AMS that automatically switches between tasks can significantly improve human performance in a multitasking scenario**. Consequently, we claim that this work has genuine implications for the design of future attention management systems, which we discuss as follows.

### 6.1 RL-Based Attention Management Demands Cognitive Models of User Behavior

The proposed concept achieved comparably high performance only with the AMS trained on the *cognitive* model, compared to the *unconstrained* model. This result was expected. The unconstrained model would, for example, be able to constantly switch back and forth within milliseconds as there are no interruption or resumption lags. The results demonstrate that a user model with simulated human constraints is essential to achieve the desired result. On average, participants yielded 1.5 times higher scores using the AMS trained with the *cognitive model* compared to their self-determined interruption strategy in the *no supervisor* condition. Still, the AMS helped less skilled players more than highly skilled players. One explanation could be that the concept itself is not robust enough to improve the performance of all high-performing humans. However, there are multiple possibilities to further improve the performance.



First, while our study used a cognitive model based on a single person, greater improvements are plausible when participants use a cognitive model fitting their individual style of playing. We plan to use approximate Bayesian computation to directly infer parameter values of participants in future studies [46, 56]. Presumably, choosing the right model might help to improve all players' performance with the AMS. Second, we could try to improve our reward function for the AMS. Third, we did not excessively use computational resources when training our agents, and longer training with improved hyperparameters could yield a better result. Fourth, we plan to improve the cognitive model itself. For example, our architecture was based on the assumption that a player would only focus on the active platform. To better account for the concurrent multitasking aspect of the game (i.e., players could always see both platforms), the *cognitive model* can be improved by an additional agent sharing the visual attention even to the currently inactive task.

## 6.2 The AMS Provoked Fast-paced Playing without Influencing Subjective Workload

The task switching behavior induced by the AMS was quite different compared to the participants' self-determined strategy. Participants spent more time on the platforms and aimed at a more stable situation before switching. This finding is in line with previous research, which suggested that humans tend to monotask until the workload in the primary task is reduced [68]. At the same time, they tended to switch to less urgent situations on the target platform than the AMS. These results can be interpreted in a way that the participants behaved more cautiously while the AMS fostered fast-paced and intensive playing. Previous research has suggested that more frequent interruptions can be beneficial over less frequent switches [54] and that faster switching may benefit rehearsal and recovery [3]. Another explanation could be that participants' load was better configured for the task [90]. One could interpret particular configurations of the source platform (small angle, little speed, and deviation of the ball) as being better suitable for task switching since such configurations are closer to "natural breakpoints" or times of "low mental load". Interestingly, participants tended to switch during such configurations, while the AMS switched at higher angles, deviations, and speeds. This would contrast with the idea that task switching always works best in such situations. Interestingly, the higher pace and the more critical situations with the AMS were not reflected in participants' subjective perception of workload. The AMS based on the *cognitive model* yielded significantly lower effort as well as mental and physical demand compared to the *no supervisor* condition. This could result from the fact that automated switching without the need for monitoring leads to less overall workload. Another explanation could be that the game was very stressful, which in turn could have provoked players to act more cautiously without support. Furthermore, it could be observed that participants with higher skills behaved more similarly to the AMS in terms of their average duration on a platform.

## 6.3 Automated Switching is Better than Interruptions in Fast-Paced Environments

Interestingly, participants performed worst in the *notification* mode, where they had to confirm task switches using a button when being interrupted, despite the suggestions originated from our best performing AMS, the *cognitive model*. Past research has suggested that self-interruptions are more disruptive than external interruptions [47]. Thus, it would be reasonable to assume that it could be easier for participants to be receptive to an interruption than to monitor both platforms and self-interrupt. However, the *notification* condition received the worst performance overall and was also subjectively rated worst in terms of the NASA-TLX scores and users' preferences. Presumably, the continuous demand of the balancing task combined with the need to confirm an upcoming interruption before acting has created an additional mental load. Another explanation could be users' tendency to defer interruptions to times of low workload or subtask boundaries even in fast-paced settings like driving [10]. Maybe participants unintentionally wanted to clean up the primary task before switching, despite this negatively influencing their performance. Research on AMSs in automated vehicles has proposed to prevent users from self-negotiating their receptivity to driving-related interruptions [84]. The results of the presented experiment confirm that automated switching has advantages over user freedom in the context of fast-paced scenarios. Still, we believe that this finding is strongly task-dependent and may differ in other, particularly less time-critical environments.

## 6.4 Adaptability to other Multitasking Scenarios

Multitasking can be categorized among various dimensions, such as the time spent on a task [67], task complexity, relation to the primary task, continuous vs. discrete tasks, and of course, the tasks' demand on perceptual, cognitive, and physical resources [13]. In this work, we have demonstrated the potential of an RL-supported AMS in a dual-task balancing game. We believe that our concept can be generalized to fast-paced, visual-motoric, and continuous multitasking situations that demand urgent reactions (i.e., driving or remote control of multiple vehicles/drones, etc.). We observed a benefit already when operating on two instances of the same task. We argue that the AMS would show an even higher performance gain with a higher number of platforms since it would be much more difficult for users to monitor all of them simultaneously. Also, our scenario allowed users to monitor both platforms concurrently. Real-life situations where this is not possible (for example, a driver focusing either on the outside environment or an in-vehicle display [43]) might even more benefit from an AMS that can utilize information outside of the users' field of view. Similarly, some multitasking situations require not only a refocus of attention and cognitive engagement but also a physical reaction (such as putting the smartphone in a safe place). Another important aspect is to investigate different tasks. In our example, we simulated a situation with two identical versions of a "primary task". We argue that the scenario did include interruption and resumption lags, as participants had to quickly accommodate another platform with a different configuration (i.e., ball speed, direction, position, etc.). Nevertheless, the effects of task switching on cognitive processes are higher when alternating between activities with

differing (perceptual, cognitive, and physical) demands and task structures [21, 30]. It will be interesting to see how the AMS is able to perform in such settings. Finally, the balancing game represents continuous tasks that progress independently from the user, where the AMS can learn that particular configurations require attention. In contrast, discrete tasks such as writing a text do not change their state while the user is engaged in another task [86]. This may require less focus of the AMS on the task environment (beyond the currently active task) and more inclusion of the users' psychophysiological states. In the future, we will (1) integrate physiological measurements to assess constructs such as stress, cognitive load, or monitoring, (2) extend our reward functions (i.e., minimizing stress levels while maximizing performance), and (3) implement a wide range of different tasks with varying demands on skill-, rule-, and knowledge-based performance (such as lane keeping in a vehicle [57], an airport control tower simulation [89], and standardized tasks like n-back [64], PASAT [78], etc.). We further call for downloading and extending our software, implementing additional tasks, or playing around with the cognitive models and algorithms to further improve performance and research of cooperative AMSs.

## 6.5 Towards General Attention Management Systems

One major downside of existing AMSs is that they are not task-independent. Most existing concepts are based on theories or operate data-driven. Theoretical assumptions require expert knowledge of the involved tasks, and data-driven approaches require large, labeled datasets [5]. In contrast, a general and task-independent AMS would be transferable to any task without specific adaptations. Computational rationality requires modeling constraints and utilities for the RL agents to reproduce human-like behavior [35]. Also our system relies on task-relevant data. We argue that given sufficient training, the AMS agent itself would not necessarily require such data as long as a suitable reward can be extracted (similar to computer games that perceive pixel streams [53]). However, the cognitive model is constrained by assumptions regarding the perception of the environment (i.e., ball positions and velocities). Potentially, there exists a set of constraints to model interruption and resumption behavior of humans independent of particular tasks, or at least with a small set of task characteristics (such as the expected physical, perceptual, or cognitive load needed to manage clean-up and resumption). Assuming optimal human performance when fully attending a task, such model could be used to train a task-independent, general AMS that minimizes interruption and resumption costs.

## 6.6 General Human Factors Considerations

Participants' subjective experiences were strongly driven by their self-rated gaming skills. Skilled players preferred the *no supervisor* condition of the AMSs. Nevertheless, the AMS based on the *cognitive model* was rated similarly in terms of the overall experience than the *no supervisor* condition. Additionally, overall subjective experience in a multitasking context depends on such characteristics as effort to play, frustration level, physical demand, and subjective performance. Consequently, the development process should not solely focus on task performance but also on other important requirements such as users' well-being. We will experiment with stress measurements as

an additional factor for our reward function in the future, with the aim of fostering both performance and relaxation. Additionally, this project focuses on human-AI cooperation in a multitasking setting while not touching upon the general question how tasks should be distributed among AI systems and humans at all (i.e., "levels of automation" [61]). Given the simplicity of the implemented task, one can argue that the best-performing system in our case would be one solely controlled by AI and without a human in the loop. We argue that such considerations require researching carefully human preferences. For example, it is known that substituting humans with technology comes with the risk of losing meaningful activities and interactions, while a proper distribution of activities can maintain user agency and the feeling of being in control [27, 49]. While answering this question is beyond the scope of the current work, we believe that RL agents operating on user models have the potential to also provide a more fine-grained and dynamic distribution of tasks in the future.

## 6.7 Limitations

Besides the issues discussed above, the following limitations must be considered. Most importantly, the scenario at hand does not represent a real-life task (beyond gaming). Both tasks (i.e., left and right platforms) are the same and have an equal state space and equal conceptual criteria like the goal state and condition-action rules. This makes it easier for humans to adapt to new situations as they change tasks compared to situations where the tasks are not equal. Therefore future work should investigate if the results of this work can also be observed for different tasks. Further, The AMS agent was trained based on a single cognitive model describing only the behavior of one participant. It can be assumed that the constraints of humans cover a wide range so a single cognitive model cannot be applied for all participants. For instance, this could be the explanation why there was no significant performance increase for some highly skilled players. For a simple task such as the one presented, a single model might have been appropriate. However, in situations where users can behave much more flexibly and have even greater variability in skills of different types (i.e., perceptual, cognitive, motor skills, etc.), personal adaptation will be essential. We also cannot rule out that participants adjusted their playing behavior based on the condition. Most likely, they sometimes also focused on the inactive platform, which was not represented in our cognitive user models. Finally, the system solely aims at productivity, which has ethical implications in a real-life context. To build successful AMSs, one also must take well-being and varying participant needs into account.

## 7 CONCLUSION

AMSs aim for better multitasking support for humans. Due to the complexity of human behavior and the involved environments, sophisticated AMSs hardly exist. In this work, we proposed to develop future systems with RL agents that utilize computational rationality. To demonstrate the capabilities of this idea, we implemented a fast-paced visual-motoric balancing game where users had to switch between two tasks to balance balls on two platforms. Our aim was to design an AMS agent that finds opportune moments for interruptions and automatically switches between the platforms. Therefore,

we used concepts of computational rationality to develop a model with cognitive constraints, which is used to train the AMS agent. The cognitive model is defined by its constraints and their extent, which are determined by means of parameters. We evaluated the approach in a user study with N=43 participants, who experienced four different conditions. We compared an AMS trained on the cognitive model with other AMS agents using an unconstrained model, as well as a notification mode where users had to confirm task switches. These conditions were compared to a baseline where participants themselves had to switch between the tasks. Thereby, we assessed performance (game scores), player/model behavior, and subjective ratings for workload. Our results show that the AMS agent trained on the cognitive model could significantly improve participants' performance, in contrast to their self-determined interruption strategies as well as the other conditions. Computational rationality was a key concept for our results as the unconstrained user model could not benefit human performance. Further, the best performing AMS using the cognitive model made participants switch between the platforms at a higher pace, while participants themselves acted more cautiously. Finally, cooperation with the AMS was rated best in terms of subjective workload according to multiple dimensions of the NASA-TLX. **With this work, we have confirmed our hypothesis that RL and CR can be used to improve human multitasking performance and build more sophisticated attention management systems in the future.** We will extend our system to include additional tasks with different user demands and re-evaluate our results. Additionally, we hope that the community uses our developed tool to research and evaluate AMSs for various purposes. Given the large amount of technology surrounding us, the time for attention management is now!

## ACKNOWLEDGMENTS

This project is supported by the Austrian Science Fund (FWF) under grant Nr.P35976-N (AITtentive) and the Research Council of Finland under grant Nrs. 328400, 341763, and 328813.

## REFERENCES

- [1] Piotr D. Adamczyk and Brian P. Bailey. 2004. If Not Now, when?: The Effects of Interruption at Different Moments Within Task Execution. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vienna, Austria) (CHI '04). ACM, New York, NY, USA, 271–278. <https://doi.org/10.1145/985692.985727>
- [2] Adam Alter. 2017. *Irresistible: The rise of addictive technology and the business of keeping us hooked*. Penguin, London, UK.
- [3] Erik M Altmann and J Gregory Trafton. 2002. Memory for goals: An activation-based model. *Cognitive science* 26, 1 (2002), 39–83.
- [4] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. 2019. Guidelines for Human-AI Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland UK) (CHI '19). Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3290605.3300233>
- [5] Christoph Anderson, Isabel Hübener, Ann-Kathrin Seipp, Sandra Ohly, Klaus David, and Veljko Pejovic. 2018. A Survey of Attention Management Systems in Ubiquitous Computing Environments. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 2, Article 58 (jul 2018), 27 pages. <https://doi.org/10.1145/3214261>
- [6] Brian P. Bailey and Joseph A. Konstan. 2006. On the need for attention-aware systems: Measuring effects of interruption on task performance, error rate, and affective state. *Computers in Human Behavior* 22, 4 (July 2006), 685–708. <https://doi.org/10.1016/j.chb.2005.12.009>
- [7] Nikola Banovic, Tovi Grossman, and George Fitzmaurice. 2013. The effect of time-based cost of error in target-directed pointing tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) (CHI '13). Association for Computing Machinery, New York, NY, USA, 1373–1382. <https://doi.org/10.1145/2470654.2466181>
- [8] Andrew G Barto, Satinder Singh, Nuttapon Chentanez, et al. 2004. Intrinsically motivated learning of hierarchical collections of skills. In *Proceedings of the 3rd International Conference on Development and Learning*, Vol. 112. Citeseer, UCSD Institute for Neural Computation, California, USA, 19.
- [9] Peter Bogunovich and Dario Salvucci. 2011. The effects of time constraints on user behavior for deferrable interruptions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (, Vancouver, BC, Canada,) (CHI '11). Association for Computing Machinery, New York, NY, USA, 3123–3126. <https://doi.org/10.1145/1978942.1979404>
- [10] Avinoam Borowsky, Noa Zangi, and Tal Oron-Gilad. 2022. Interruption management in the context of take-over-requests in conditional driving automation. *IEEE Transactions on Human-Machine Systems* 52, 5 (2022), 1015–1024.
- [11] JP Borst and NA Taatgen. 2007. The costs of multitasking in threaded cognition. In *Proceedings of the eighth international conference on cognitive modeling*. Psychology Press, New York, USA, 133–138.
- [12] Jelmer P Borst, Niels A Taatgen, and Hedderik Van Rijn. 2010. The problem state: a cognitive bottleneck in multitasking. *Journal of Experimental Psychology: Learning, memory, and cognition* 36, 2 (2010), 363.
- [13] Jelmer P. Borst, Niels A. Taatgen, and Hedderik van Rijn. 2015. What Makes Interruptions Disruptive? A Process-Model Account of the Effects of the Problem State Bottleneck on Task Interruption and Resumption. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) (CHI '15). Association for Computing Machinery, New York, NY, USA, 2971–2980. <https://doi.org/10.1145/2702123.2702156>
- [14] Duncan P. Brumby, Christian P. Janssen, Tuomo Kujala, and Dario D. Salvucci. 2018. Computational Models of User Multitasking. In *Computational Interaction*. Oxford University Press, Oxford, England. <https://doi.org/10.1093/oso/9780198799603.003.0013> arXiv:<https://academic.oup.com/book/0/chapter/297822125/chapter-pdf/40117570/oso-9780198799603-chapter-13.pdf>
- [15] Andreas Bulling. 2016. Pervasive Attentive User Interfaces. *Computer* 49, 1 (Jan. 2016), 94–98. <https://doi.org/10.1109/MC.2016.32>
- [16] David M Cades, Deborah A Boehm Davis, J Gregory Trafton, and Christopher A Monk. 2007. Does the difficulty of an interruption affect our ability to resume?. In *Proceedings of the human factors and ergonomics society annual meeting*, Vol. 51. SAGE Publications Sage CA, Los Angeles, CA, 234–238.
- [17] David M Cades, Deborah A Boehm Davis, J Gregory Trafton, and Christopher A Monk. 2007. Does the difficulty of an interruption affect our ability to resume?. In *Proceedings of the human factors and ergonomics society annual meeting*, Vol. 51. SAGE Publications Sage CA, Los Angeles, CA, 234–238.
- [18] Xiuli Chen, Gilles Bailly, Duncan P. Brumby, Antti Oulasvirta, and Andrew Howes. 2015. The Emergence of Interactive Behavior: A Model of Rational Menu Search. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) (CHI '15). Association for Computing Machinery, New York, NY, USA, 4217–4226. <https://doi.org/10.1145/2702123.2702483>
- [19] Xiuli Chen, Sandra Dorothee Starke, Chris Baber, and Andrew Howes. 2017. A Cognitive Model of How People Make Decisions Through Interaction with Visual Displays. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) (CHI '17). Association for Computing Machinery, New York, NY, USA, 1205–1216. <https://doi.org/10.1145/3025453.3025596>
- [20] Driving Automation Systems Committee et al. 2016. *Human Factors Definitions for Automated Driving and Related Research Topics*. SAE International, Pennsylvania, USA.
- [21] Mary Czerwinski, Edward Cutrell, and Eric Horvitz. 2000. Instant messaging: Effects of relevance and timing. In *People and computers XIV: Proceedings of HCI*, Vol. 2. Springer, London, UK, 71–76.
- [22] Justin Edwards, Christian Janssen, Sandy Gould, and Benjamin R. Cowan. 2021. Eliciting Spoken Interruptions to Inform Proactive Speech Agent Design. In *Proceedings of the 3rd Conference on Conversational User Interfaces* (Bilbao (online), Spain) (CUI '21). Association for Computing Machinery, New York, NY, USA, Article 23, 12 pages. <https://doi.org/10.1145/3469595.3469618>
- [23] C Ferreira and M Geig. 2018. Get Started with the Unity\* Entity Component System (ECS), C# Job System, and Burst Compiler. *Intel Corporation, [Online]*. Available: <https://www.intel.com/content/www/us/en/developer/articles/guide/get-started-with-the-unity-entity-component-system-ecs-c-sharp-job-system-and-burst-compiler.html> (visited on 2022-11-12) 1 (2018), 1.
- [24] Joel E. Fischer, Chris Greenhalgh, and Steve Benford. 2011. Investigating Episodes of Mobile Phone Activity as Indicators of Opportune Moments to Deliver Notifications. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services* (Stockholm, Sweden) (Mobile-HCI '11). Association for Computing Machinery, New York, NY, USA, 181–190. <https://doi.org/10.1145/2037373.2037402>
- [25] Philip J. Fleming and John J. Wallace. 1986. How Not to Lie with Statistics: The Correct Way to Summarize Benchmark Results. *Commun. ACM* 29, 3 (mar 1986), 218–221. <https://doi.org/10.1145/5666.5673>

- [26] Thomas Franke, Christiane Attig, and Daniel Wessel. 2019. A personal resource for technology interaction: development and validation of the affinity for technology interaction (ATT) scale. *International Journal of Human-Computer Interaction* 35, 6 (2019), 456–467.
- [27] Anna-Katharina Frison, Philipp Wintersberger, Andreas Riener, and Clemens Schartmüller. 2017. Driving Hotzenplotz: A Hybrid Interface for Vehicle Control Aiming to Maximize Pleasure in Highway Driving. In *Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* (Oldenburg, Germany) (*AutomotiveUI '17*). ACM, New York, NY, USA, 236–244. <https://doi.org/10.1145/3122986.3123016>
- [28] Christoph Gebhardt, Brian Hecox, Bas van Opheusden, Daniel Wigdor, James Hillis, Otmar Hilliges, and Hrvoje Benko. 2019. Learning Cooperative Personalized Policies from Gaze Data. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) (*UIST '19*). Association for Computing Machinery, New York, NY, USA, 197–208. <https://doi.org/10.1145/3332165.3347933>
- [29] Christoph Gebhardt, Antti Oulasvirta, and Otmar Hilliges. 2020. Hierarchical reinforcement learning as a model of human task interleaving. *Computational Brain & Behavior* 4 (2020), 284–304.
- [30] Sandy JJ Gould, Duncan P Brumby, and Anna L Cox. 2013. What does it mean for an interruption to be relevant? An investigation of relevance as a memory effect. In *Proceedings of the human factors and ergonomics society annual meeting*, Vol. 57. SAGE Publications Sage CA, Los Angeles, CA, USA, 149–153.
- [31] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Human Mental Workload*, Peter A. Hancock and Najmedin Meshkati (Eds.). Advances in Psychology, Vol. 52. North-Holland, Amsterdam, Netherlands, 139–183. [https://doi.org/10.1016/S0166-4115\(08\)62386-9](https://doi.org/10.1016/S0166-4115(08)62386-9)
- [32] Helen M Hodgetts and Dylan M Jones. 2006. Interruption of the Tower of London task: support for a goal-activation approach. *Journal of Experimental Psychology: General* 135, 1 (2006), 103.
- [33] Eric Horvitz. 1999. Principles of Mixed-initiative User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Pittsburgh, Pennsylvania, USA) (*CHI '99*). ACM, New York, NY, USA, 159–166. <https://doi.org/10.1145/302979.303030>
- [34] ECMCE Horvitz. 2001. Notification, disruption, and memory: Effects of messaging interruptions on memory and performance. In *Human-Computer Interaction: INTERACT*, Vol. 1. IOS Press, Amsterdam, Netherlands, 263.
- [35] Andrew Howes, Jussi P. Jokinen, and Antti Oulasvirta. 2023. Towards machines that understand people. *AI Magazine* 44, 3 (2023), 312–327.
- [36] Shamsi T. Iqbal and Brian P. Bailey. 2005. Investigating the Effectiveness of Mental Workload As a Predictor of Opportune Moments for Interruption. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems* (Portland, OR, USA) (*CHI EA '05*). ACM, New York, NY, USA, 1489–1492. <https://doi.org/10.1145/1056808.1056948>
- [37] Shamsi T. Iqbal and Brian P. Bailey. 2008. Effects of intelligent notification management on users and their tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) (*CHI '08*). Association for Computing Machinery, New York, NY, USA, 93–102. <https://doi.org/10.1145/1357054.1357070>
- [38] Shamsi T. Iqbal and Brian P. Bailey. 2010. Oasis: A framework for linking notification delivery to the perceptual structure of goal-directed tasks. *ACM Transactions on Computer-Human Interaction (TOCHI)* 17, 4 (2010), 1–28.
- [39] Shamsi T. Iqbal and Eric Horvitz. 2010. Notifications and awareness: a field study of alert usage and preferences. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work* (Savannah, Georgia, USA) (*CSCW '10*). Association for Computing Machinery, New York, NY, USA, 27–30. <https://doi.org/10.1145/1718918.1718926>
- [40] Christian P. Janssen, Sandy J.J. Gould, Simon Y.W. Li, Duncan P. Brumby, and Anna L. Cox. 2015. Integrating knowledge of multitasking and interruptions across different perspectives and research methods. *International Journal of Human-Computer Studies* 79 (2015), 1–5. <https://doi.org/10.1016/j.ijhcs.2015.03.002>
- [41] JASP Team. 2023. JASP (Version 0.18.0)[Computer software]. <https://jasp-stats.org/>
- [42] Jussi Jokinen, Aditya Acharya, Mohammad Uzair, Xinhui Jiang, and Antti Oulasvirta. 2021. Touchscreen Typing As Optimal Supervisory Control. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (, Yokohama, Japan,) (*CHI '21*). Association for Computing Machinery, New York, NY, USA, Article 720, 14 pages. <https://doi.org/10.1145/3411764.3445483>
- [43] Jussi P. Jokinen, Tuomo Kujala, and Antti Oulasvirta. 2021. Multitasking in driving as optimal adaptation under uncertainty. *Human factors* 63, 8 (2021), 1324–1341.
- [44] Jussi P. P. Jokinen and Tuomo Kujala. 2021. Modelling Drivers' Adaptation to Assistance Systems. In *13th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* (Leeds, United Kingdom) (*AutomotiveUI '21*). Association for Computing Machinery, New York, NY, USA, 12–19. <https://doi.org/10.1145/3409118.3475150>
- [45] Arthur Juliani, Vincent-Pierre Berges, Ervin Teng, Andrew Cohen, Jonathan Harper, Chris Elion, Chris Goy, Yuan Gao, Hunter Henry, and Marwan Matrar. 2018. Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627* 1 (2018), 28.
- [46] Antti Kangasrääsiö, Kumaripaba Athukorala, Andrew Howes, Jukka Corander, Samuel Kaski, and Antti Oulasvirta. 2017. Inferring Cognitive Models from Data using Approximate Bayesian Computation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) (*CHI '17*). Association for Computing Machinery, New York, NY, USA, 1295–1306. <https://doi.org/10.1145/3025453.3025576>
- [47] Ioanna Katidioti, Jelmer P. Borst, Marieke K. van Vugt, and Niels A. Taatgen. 2016. Interrupt me: External interruptions are less disruptive than self-interruptions. *Computers in Human Behavior* 63 (2016), 906–915.
- [48] Harmanpreet Kaur, Alex C. Williams, Daniel McDuff, Mary Czerwinski, Jaime Teevan, and Shamsi T. Iqbal. 2020. Optimizing for Happiness and Productivity: Modeling Opportune Moments for Transitions and Breaks at Work. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (, Honolulu, HI, USA,) (*CHI '20*). Association for Computing Machinery, New York, NY, USA, 1–15. <https://doi.org/10.1145/3313831.3376817>
- [49] Holger Klapperich and Marc Hassenzahl. 2016. Hotzenplotz: Reconciling Automation with Experience. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction* (Gothenburg, Sweden) (*NordiCHI '16*). Association for Computing Machinery, New York, NY, USA, Article 39, 10 pages. <https://doi.org/10.1145/2971485.2971532>
- [50] Joerg Kubitzki and Wolfgang Fastenmeier. 2016. *Ablenkung durch moderne Informations- und Kommunikationstechniken und soziale Interaktion bei Autofahrern*. Allianz AG, Stuttgart, Germany.
- [51] Peng Liao, Kristjan Greenewald, Predrag Klasnja, and Susan Murphy. 2020. Personalized heartsteps: A reinforcement learning algorithm for optimizing physical activity. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 1–22.
- [52] Gloria Mark, Daniela Gudith, and Ulrich Klocke. 2008. The cost of interrupted work: more speed and stress. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) (*CHI '08*). Association for Computing Machinery, New York, NY, USA, 107–110. <https://doi.org/10.1145/1357054.1357072>
- [53] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. *arXiv:1312.5602 [cs.LG]*
- [54] Christopher A. Monk. 2004. The effect of frequent versus infrequent interruptions on primary task resumption. In *Proceedings of the human factors and ergonomics society annual meeting*, Vol. 48. SAGE Publications Sage CA, Los Angeles, CA, USA, 295–299.
- [55] Christopher A. Monk, J. Gregory Trafton, and Deborah A. Boehm-Davis. 2008. The effect of interruption duration and demand on resuming suspended goals. *Journal of experimental psychology: Applied* 14, 4 (2008), 299.
- [56] Hee-Seung Moon, Antti Oulasvirta, and Byungjoo Lee. 2023. Amortized Inference with User Simulations. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (, Hamburg, Germany,) (*CHI '23*). Association for Computing Machinery, New York, NY, USA, Article 773, 20 pages. <https://doi.org/10.1145/3544548.3581439>
- [57] Menno Nijboer, Jelmer P. Borst, Hedderik Van Rijn, and Niels A. Taatgen. 2016. Driving and multitasking: the good, the bad, and the dangerous. *Frontiers in psychology* 7 (2016), 1718.
- [58] Tadashi Okoshi, Julian Ramos, Hiroki Nozaki, Jin Nakazawa, Anind K. Dey, and Hideyuki Tokuda. 2015. Reducing users' perceived mental effort due to interruptive notifications in multi-device mobile environments. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Osaka, Japan) (*UbiComp '15*). Association for Computing Machinery, New York, NY, USA, 475–486. <https://doi.org/10.1145/2750858.2807517>
- [59] Antti Oulasvirta, Jussi P. P. Jokinen, and Andrew Howes. 2022. Computational Rationality as a Theory of Interaction. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (, New Orleans, LA, USA,) (*CHI '22*). Association for Computing Machinery, New York, NY, USA, Article 359, 14 pages. <https://doi.org/10.1145/3491102.3517739>
- [60] Antti Oulasvirta and Pertti Saariluoma. 2006. Surviving task interruptions: Investigating the implications of long-term working memory theory. *International Journal of Human-Computer Studies* 64, 10 (Oct. 2006), 941–961. <https://doi.org/10.1016/j.ijhcs.2006.04.006>
- [61] Raja Parasuraman, Thomas B. Sheridan, and Christopher D. Wickens. 2000. A model for types and levels of human interaction with automation. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans* 30, 3 (2000), 286–297.
- [62] Tomi Peltola, Jussi Jokinen, and Samuel Kaski. 2019. Probabilistic Formulation of the Take The Best Heuristic. *arXiv:1911.00572 [stat.AP]*
- [63] Benjamin Poppinga, Wilko Heuten, and Susanne Boll. 2014. Sensor-Based Identification of Opportune Moments for Triggering Notifications. *IEEE Pervasive Computing* 13, 1 (2014), 22–29. <https://doi.org/10.1109/MPRV.2014.15>

- [64] Brandon CW Ralph and Daniel Smilek. 2017. Individual differences in media multitasking and performance on the n-back. *Attention, Perception, & Psychophysics* 79 (2017), 582–592.
- [65] Jens Rasmussen. 1983. Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man, and Cybernetics* SMC-13, 3 (1983), 257–266. <https://doi.org/10.1109/TSMC.1983.6313160>
- [66] Claudia Roda and Julie Thomas. 2006. Attention aware systems: Theories, applications, and research agenda. *Computers in Human Behavior* 22, 4 (2006), 557–587.
- [67] Dario D. Salvucci and Peter Bogunovich. 2010. Multitasking and monotasking: the effects of mental workload on deferred task interruptions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) (CHI '10). Association for Computing Machinery, New York, NY, USA, 85–88. <https://doi.org/10.1145/1753326.1753340>
- [68] Dario D Salvucci, Christopher A Monk, and J Gregory Trafton. 2009. A process-model account of task interruption and resumption: When does encoding of the problem state occur?. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 53. SAGE Publications Sage CA, Los Angeles, CA, USA, 799–803.
- [69] Dario D Salvucci and Niels A Taatgen. 2008. Threaded cognition: an integrated theory of concurrent multitasking. *Psychological review* 115, 1 (2008), 101.
- [70] Dario D. Salvucci, Niels A. Taatgen, and Jelmer P. Borst. 2009. Toward a unified theory of the multitasking continuum: from concurrent performance to task switching, interruption, and resumption. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) (CHI '09). Association for Computing Machinery, New York, NY, USA, 1819–1828. <https://doi.org/10.1145/1518701.1518981>
- [71] Jeff Sauro. 2010. A practical guide to measuring usability. *Measuring Usability LLC, Denver* 12 (2010), 124.
- [72] Christina Schneegass and Fiona Draxler. 2021. Designing Task Resumption Cues for Interruptions in Mobile Learning Scenarios. In *Technology-Augmented Perception and Cognition*, Evangelos Niforatos and Tilman Dingler (Eds.). Springer International Publishing, Cham, 125–181. [https://doi.org/10.1007/978-3-030-30457-7\\_5](https://doi.org/10.1007/978-3-030-30457-7_5) Series Title: Human–Computer Interaction Series.
- [73] Siti Rohkmah Mohd Shukri and Andrew Howes. 2019. Children adapt drawing actions to their own motor variability and to the motivational context of action. *International Journal of Human-Computer Studies* 130 (2019), 152–165.
- [74] Cheri Speier, Joseph S Valacich, and Iris Vessey. 1999. The influence of task interruption on individual decision making: An information overload perspective. *Decision sciences* 30, 2 (1999), 337–360.
- [75] Namrata Srivastava, Rajiv Jain, Jennifer Healey, Zoya Bylinskii, and Tilman Dingler. 2021. Mitigating the Effects of Reading Interruptions by Providing Reviews and Previews. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) (CHI EA '21). Association for Computing Machinery, New York, NY, USA, Article 229, 6 pages. <https://doi.org/10.1145/3411763.3451610>
- [76] C. Steinhurst. 2020. How To Reclaim The Huge Losses That Multitasking Forces On Your Company. *Forbes.com*. Retrieved August 22, 2020 from <https://www.forbes.com/sites/curtsteinhurst/2020/02/28/how-to-reclaim-the-huge-losses-that-multitasking-forces-on-your-company/?sh=26d7e8cbc024>.
- [77] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press, Cambridge, MA.
- [78] Tom N Tombaugh. 2006. A comprehensive review of the paced auditory serial addition test (PASAT). *Archives of clinical neuropsychology* 21, 1 (2006), 53–76.
- [79] J Gregory Trafton and Christopher A Monk. 2007. Task interruptions. *Reviews of human factors and ergonomics* 3, 1 (2007), 111–126.
- [80] Roel Vertegaal et al. 2003. Attentive user interfaces. *Commun. ACM* 46, 3 (2003), 30–33.
- [81] Peter Vorderer, Dorothee Hefner, Leonard Reinecke, and Christoph Klimmt. 2017. *Permanently online, permanently connected: Living and communicating in a POPC world*. Routledge, New York; London.
- [82] Mark Weiser. 1991. The Computer for the 21 st Century. *Scientific american* 265, 3 (1991), 94–105.
- [83] Christopher D Wickens. 2020. Processing resources and attention. In *Multiple task performance*. CRC Press, Florida, USA, 3–34.
- [84] Philipp Wintersberger, Andreas Riener, Clemens Schartmüller, Anna-Katharina Frison, and Klemens Weigl. 2018. Let Me Finish before I Take Over: Towards Attention Aware Device Integration in Highly Automated Vehicles. In *Proceedings of the 10th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* (Toronto, ON, Canada) (AutomotiveUI '18). Association for Computing Machinery, New York, NY, USA, 53–65. <https://doi.org/10.1145/3239060.3239085>
- [85] Philipp Wintersberger, Clemens Schartmüller, and Andreas Riener. 2019. Attentive user interfaces to improve multitasking and take-over performance in automated driving: the auto-net of things. *International Journal of Mobile Human Computer Interaction (IJMHCI)* 11, 3 (2019), 40–58.
- [86] Philipp Wintersberger, Niels van Berkel, Nadia Fereydooni, Benjamin Tag, Elena L. Glassman, Daniel Buschek, Ann Blandford, and Florian Michahelles. 2022. Designing for Continuous Interaction with Artificial Intelligence Systems. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) (CHI EA '22). Association for Computing Machinery, New York, NY, USA, Article 170, 4 pages. <https://doi.org/10.1145/3491101.3516409>
- [87] Glenn Wylie and Alan Allport. 2000. Task switching and the measurement of “switch costs”. *Psychological research* 63, 3 (2000), 212–233.
- [88] Songlin Xu and Xinyu Zhang. 2023. Augmenting Human Cognition with an AI-Mediated Intelligent Visual Feedback. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (CHI '23). Association for Computing Machinery, New York, NY, USA, Article 26, 16 pages. <https://doi.org/10.1145/3544548.3580905>
- [89] Birsan Yörük Açikel, Uğur Turhan, and Yavuz Akbulut. 2018. Effect of multitasking on simulator sickness and performance in 3D aerodrome control training. *Simulation & Gaming* 49, 1 (2018), 27–49.
- [90] Mark S Young and Neville A Stanton. 2002. Attention and automation: New perspectives on mental underload and performance. *Theoretical issues in ergonomics science* 3, 2 (2002), 178–194.
- [91] Difeng Yu, Ruta Desai, Ting Zhang, Hrvoje Benko, Tanya R. Jonker, and Aakar Gupta. 2022. Optimizing the Timing of Intelligent Suggestion in Virtual Reality. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) (UIST '22). Association for Computing Machinery, New York, NY, USA, Article 6, 20 pages. <https://doi.org/10.1145/3526113.3545632>
- [92] Manuela Züger, Christopher Corley, André N. Meyer, Boyang Li, Thomas Fritz, David Shepherd, Vinay Augustine, Patrick Francis, Nicholas Kraft, and Will Snipes. 2017. Reducing Interruptions at Work: A Large-Scale Field Study of FlowLight. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) (CHI '17). Association for Computing Machinery, New York, NY, USA, 61–72. <https://doi.org/10.1145/3025453.3025662>
- [93] Manuela Züger, Sebastian C. Müller, André N. Meyer, and Thomas Fritz. 2018. Sensing Interruptibility in the Office: A Field Study on the Use of Biometric and Computer Interaction Sensors. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (CHI '18). Association for Computing Machinery, New York, NY, USA, 1–14. <https://doi.org/10.1145/3173574.3174165>