

Do humans adapt their planning horizon?

An analysis of sequential decision-making in the videogame Frogger

Bachelor thesis by Tobias Fabian Niehues

Date of submission: October 27, 2022

1. Review: Prof. Constantin Rothkopf, PhD
2. Review: Florian Kadner, M.Sc.
Darmstadt



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Human Sciences
Department

Psychology of Information
Processing (PIP)

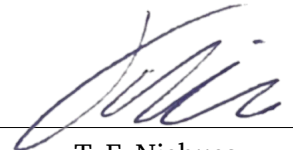
Erklärung zur Abschlussarbeit gemäß § 22 Abs. 7 APB TU Darmstadt

Hiermit versichere ich, Tobias Fabian Niehues, die vorliegende Bachelorarbeit gemäß § 22 Abs. 7 APB der TU Darmstadt ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Falle eines Plagiats (§ 38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Darmstadt, 27. Oktober 2022



T. F. Niehues

Abstract

Humans can employ sophisticated strategies to plan their next actions while only having limited cognitive capacity. Most of the studies investigating human behavior focus on minimal and rather abstract tasks. We provide an environment inspired by the video game *Frogger*, especially designed for studying how far humans plan ahead. This is described by the planning horizon, a not directly measurable quantity representing how far ahead people can consider the consequences of their actions. We treat the subjects' eye movements as externalization of their internal planning horizon and can thereby infer its development over time.

We found that people can dynamically adapt their planning horizon when switching between tasks. While subjects employed bigger planning horizons, we could not measure any physiological indicators of stronger cognitive engagement. Subjects using larger planning horizons were able to score higher in the game. We designed neural networks for predicting the subject's planning horizon in different situations, providing further insight on the key features needed for accurate predictions of the planning horizon. In general, models trained only on subject-specific data achieved higher accuracy than models trained on data collected from all subjects.

Contents

1. Introduction	6
2. Fundamentals	8
2.1. Definition of planning	8
2.2. Observability of planning processes	9
2.3. Characteristics of the human planning horizon	10
2.3.1. Myopic versus long-term planning	10
2.3.2. Planning horizon as a dynamic quantity	10
2.4. Video games for investigating the human planning horizon	11
3. Experiment design	13
3.1. Environment	13
3.1.1. Basic structure	13
3.1.2. Differences to vanilla Frogger	17
3.2. Eye tracker and hardware	18
3.3. Experiment design	18
3.3.1. Participants	18
4. Analysis	19
4.1. Detection of eye movement events	19
4.1.1. Types of eye movements	19
4.1.2. Algorithmic implementation	19
4.1.3. Blink detection	21
4.1.4. Measuring pupil size	22
4.2. Accounting for different fixation durations	23
5. Experiment results	24
5.1. Descriptive statistics	24
5.1.1. Performance	24
5.1.2. Movements / Behavior	27
5.1.3. Gaze	30
5.2. River versus street	31
5.2.1. Mean Fixation Distance (MFD)	32
5.2.2. Direction of fixations	33
5.2.3. Blinking behavior	35
5.2.4. Pupil size	36
5.3. Experts versus novices	37
5.3.1. Mean Fixation Distance (MFD)	37
5.3.2. Blinking behavior	38

5.3.3. Pupil size	38
6. Gaze prediction by neural networks	40
6.1. Input / output data	40
6.2. Architectures	43
6.2.1. Fully-connected	43
6.2.2. Convolutional	44
6.2.3. Recurrent / Long-Short-Term Memory	45
6.2.4. Recurrent convolutional	46
6.3. Training process	47
6.4. Results	49
6.4.1. Overview	49
6.4.2. Results over all subjects	49
6.4.3. Results for subject-specific training	52
7. Discussion	55
7.1. Experiment results	55
7.1.1. Humans adapt their planning horizon dynamically when switching tasks	55
7.1.2. High-performing subjects employ increased planning horizons	56
7.1.3. Task-specificity of the environment	57
7.2. Modeling with neural networks	57
7.2.1. Performance differences between architectures	58
7.2.2. Neural networks perform better on single-subject data	58
7.2.3. Insights from neural network results	58
7.2.4. Homogeneity and structure of data	59
7.3. Future work	60
7.3.1. Experiment variants	60
7.3.2. Extending experiments with neural networks	61
8. Conclusion	64
A. Appendix	70

1. Introduction

Humans are capable of performing well in a plethora of different tasks, despite their limited cognitive capacity. The key to this success is their ability to *plan* their next actions. That is, people are able to virtually simulate the outcomes of different sequences of actions and by that select an action with highest probability to result in the favored outcome. We can observe this skill directly when people are planning their route to travel from A to B, but it is also present in any other kind of task. For example, in social conversation people plan what to say next to steer the course of the conversation and evoke certain reactions in their vis-à-vis or even plan different trajectories for apparently simple movements to grasp a difficult to reach object.

Since humans' cognitive skills are computationally finite (Callaway et al., 2022b), the question remains which information people assign the most relevance to and how they allocate computational resources to efficiently make the most use of this (Griffiths et al., 2019). This also covers how people are able to plan sufficiently far ahead and how long this *planning horizon* actually reaches. Research regarding the human planning horizon is essential for predicting and understanding human behavior overall, since it is omnipresent in every action people take. While some studies assume rather myopic planning horizons, recent research could find evidence for longer planning horizons that also allow for better performance in tasks (Ma et al., 2021). Other studies show that people resort to shorter planning horizons when situations take an unexpected course (Carton et al., 2016). With further advances in the domain of human decision-making and planning behavior, it is also possible to teach people how to plan optimally, for instance supported by artificial intelligence (Callaway et al., 2022a).

Furthermore, advances in this domain can help leverage mechanisms of the human mind in artificial intelligence. For instance, in reinforcement learning settings agents learn certain behaviors by maximizing the gain of a sequence of rewarded and penalized actions, where the agents usually can only consider sequences of finite length. A well performing example of this is the EMPA-model by Tsvidis et al. (2021), which reached human-level performance by employing mechanisms which mimic a dynamic planning horizon. The success of several other machine learning models, like *AlphaGo*, is based on efficient planning, which employs pruning a tree of simulated consequences of various actions to the most relevant information and thereby virtually decreasing the planning horizon (Silver et al., 2016).

Most of the research from cognitive science and psychology regarding human behavior in various tasks is focused on very basic, abstract tasks, where human behavior can be examined easily. In experiments from the domain of artificial intelligence, research usually refers to either rather complex tasks to present the superior performance of their models or focuses on simple tasks that are especially difficult to solve for artificial intelligence but not humans (van Opheusden and Ma, 2019). This results in a lack of studies on human behavior in tasks in more realistic environments. Video games provide a good basis for less abstract and more natural tasks, while still providing controllable and low-scale environments. For example, Şimşek et al. (2016) and Wang et al. (2020) investigated human behavior and decision-making in video games and similar environments.

For that purpose, we designed an experiment inspired by the video game *Frogger*, presenting the subjects with a small amount of basic tasks and providing possibilities for a lot of different movement trajectories to solve the tasks successfully. Thus, the subjects could exhaust their planning abilities. By tracking their

eye movements, we could measure the subjects' planning horizons throughout each trial, by treating their fixations as externalized estimates of their current planning horizon (Yang et al., 2016; Zhu et al., 2022). Furthermore, we designed various neural networks with different architectures and trained them on the collected data. This allowed us to make new assumptions on the key features needed to predict the human planning horizon and how to improve architectures for this area of application.

2. Fundamentals

The following section introduces a definition of planning and the most important concepts underlying human planning strategies in section 2.1. Section 2.2 explains how internal cognitive processes can be monitored indirectly from the outside, followed by section 2.3 gives a short introduction about the basic characteristics underlying human planning processes as a cognitive quantity. This chapter closes with section 2.4, providing a short excursion to the domain of video games and their relevance in research, especially regarding human behavior.

2.1. Definition of planning

All living beings are forced to make several decisions each minute of the day, be it to find a suitable trajectory for executing movements, when and what to eat, thinking about their long-time goals in life and how to achieve them or any other situation where several actions can be taken. Usually, the course these situations take depends on the choice of action and the advantages and disadvantages this action brings with it. Thus, already from an evolutionary perspective animals and humans must have developed mechanisms to effectively make the right choices. In earlier days, this often might have been essential for survival.

The fastest and simplest way for handling decisions that require an optimal selection of actions is formed by habitual reactions. With such habit-based control, all living creatures had prior experience with certain situations and some actions have proven to be more useful than others. Making use of such habits, there is **no** deliberate consideration of the outcomes different actions might result in. It is simply the dominant action that yielded the best outcomes in this and similar situations previously. (Dolan and Dayan, 2013; Redish, 2016)

Contrary to this habitual behavior, there is what we call *planning*. That is, the explicit and deliberate consideration of the consequences of different actions. In contemplating about what might be the best action to take in a certain situation, we picture which following situations might arise when we take a certain action. That is, we use our knowledge of the environment and its current state to virtually simulate the course of events. This is done for each case in which one of all the possible actions is taken. Herein, planning creates new information about the different actions from the knowledge we already have about the environment. This knowledge could originate from, for instance, previous experiences in the same or similar situations. This is an ability that only mammals and birds seem to have (Doll et al., 2015; Jones et al., 2012; Schmidt et al., 2019), which might be due to an increase in visual information and more complex environment structures on land in contrast to water (Mugan and MacIver, 2020; MacIver et al., 2017). This clearly is a more advanced approach on how to make best use of the resources and possibilities one has, but is also potentially more time-consuming and more computationally demanding.

The simulation of consequences for all possible actions is computationally very intensive, though humans are constrained in their cognitive abilities by their computational resources and memory capacities. Therefore, the question on how humans allocate resources to be able to make choices with long-term consequences (Griffiths et al., 2019) and if and what heuristics they might use arises naturally (Callaway et al., 2022b).

Exponential character of planning

Because of these computational limits, we are not able to simulate the outcomes of each action in each situation and their consequences infinitely. Especially, since the space of all possible consequences extends exponentially.

Consider the case that we are in a certain situation, in which we can take one of two possible actions. We will subsequently refer to these situations as *states* and to ‘us’ as *agents*, in line with the technical terms used in any literature to decision-making and artificial intelligence. If each action results in another state in which the agent has the choice over two actions again and this pattern continues, the number of states the agent needs to simulate is 2^N , where N denotes the number of time steps we plan ahead. This N denotes the so-called *planning horizon* and must be finite since we do not have infinite computational capacities. The new question arising now is, how far are humans able to plan ahead or, more concrete, how big is this planning horizon for humans exactly and how are we able to measure it?

2.2. Observability of planning processes

Essentially, planning is an internal mechanism for generating new information about the consequences of the actions we might take. Because we only virtually simulate the outcomes of the states by considering different actions in our minds, this is a completely internal process. On the contrary, we could employ a trial-and-error approach trying out different actions. While this trial-and-error approach is an external process, we are able to observe it completely because we actually see which actions an agent tries to take. This is not the case for any planning processes.

Still, to actually make use of the planning abilities humans have, we need to have information about the current state of the environment we are in. If we are planning a route to a city using different highways but only check the next section for any traffic news, we cannot create an adequate plan for avoiding traffic jams on the last sections of the route. Since we are acting in and interacting with a physical world, we need to make use of our senses to sample information from our surroundings. From this sensory information we can infer the current state of the environment around us and use this knowledge in planning and simulating the future development of the environment’s state as well.

Since the visual sensory system is our dominant and primary sense, humans will mostly rely on visual information to infer the current environment state. This means, that agents will direct their attention and gaze towards parts of the environment where they do not know or are uncertain of the current state to maximize the information gain of each gaze fixation (Hayhoe and Ballard, 2005; Callaway et al., 2021; Yang et al., 2016). They will not be able to include parts of the environment state into their planning that they did not scan. That is, the eye movements and gaze of the agents is closely related to their planning horizon, since the planning horizon clearly relies on this so-called *information sampling* on the environment (Ma et al., 2021). The eye movements ‘externalize’ the cognitive process of planning and by tracking the agents’ eye movements, we can thus infer the size of their current planning horizon (Zhu et al., 2022). If the agents only look at the

environments in their near vicinity, they will probably have a small planning horizon and vice versa. This offers a great opportunity for making quantitative inferences about people's planning horizon without needing to rely on think-aloud protocols or other subject-made protocols, that are usually quite error prone since introspection often does not resemble the subjects' true inner cognitive states (Schwitzgebel, 2008).

2.3. Characteristics of the human planning horizon

The size of the human planning horizon may be biased towards longer or shorter horizons. Also, since people are able to employ various different planning strategies, the planning horizon might also be a dynamic and not static quantity. The certain characteristics of human planning horizons will be explored in this section.

2.3.1. Myopic versus long-term planning

Recent studies show that people do not plan ahead as far as possible or needed to find an optimal strategy and thus seem to have a bias for myopic planning (Meder et al., 2019). This is, for instance, the case when people only consider the near future in the course of events or direct most part of the areas fixated by their eyes towards their immediate vicinity in tasks where relevant information is distributed spatially. Still, other studies found that planning is not necessarily myopic and observed longer planning horizons in their tasks than previous works assumed (Ma et al., 2021). Also, Callaway et al. (2021) found that human planning strategies are in line with the concept of optimal information seeking.

While there are several studies suggesting different lengths of planning horizons, we must keep in mind that there is not one 'true' length for the planning horizon that will always be employed by all people. People have inter-individual differences and will also employ different planning strategies in different environments and tasks (Jain et al., 2022), that might also substantially differ in their corresponding planning horizons. For example, Jain et al. (2022) found 79 different strategies that could be grouped into 13 groups by similarity (e.g. forward planning, backward planning, breadth-first search, ...) for different environments, who were solved by 164 participants. People might even change their planning strategy within the same task multiple times (Lee et al., 2019).

2.3.2. Planning horizon as a dynamic quantity

Since people employ a variety of different strategies even over the same task, the planning horizon will also not be a static quantity that always stays the same, since some tasks or environments might need more or less planning ahead, and thus bigger or smaller planning horizons. As soon as we accept the planning horizon as well as the underlying planning strategies as dynamic quantities that may change anytime, we can start looking for rules that might govern the switching of planning strategies and horizons.

Carton et al. (2016) found that people employ shorter planning horizons in obstacle-avoidance tasks as soon as an object moves into the course of their already planned trajectory. This makes sense, since people need to adapt to the new situation promptly and thus cannot spend a lot of time in computing the optimal plan regarding its long-term consequences. The concept of dynamic planning horizons that might change depending on the current state of the environment is also supported by Tsvividis et al. (2021), which were able

to create a model for reinforcement learning that can also capture and replicate human behavior by employing different planning horizons in its algorithms in different situations.

As an example from the domain of artificial intelligence, the great success of DeepMind's *AlphaGo* was also due to pruning the decision tree in the algorithm which plans further ahead and simulates the course of the game when taking different actions (Silver et al., 2016). Since this mechanism strives to keep the demands for computing power low to allow for real-time computations, similar mechanism might also be part of human cognitive processes. This also suggests that a dynamic planning horizon with varying length is preferable to a static planning horizon.

2.4. Video games for investigating the human planning horizon

Section 2.3.2 also introduced that people are able to employ various different strategies when it comes to planning. To investigate the human planning horizon, we need find tasks that can be solved in a controlled environment to allow for the same conditions for every participants, while also not distorting the results because of unnatural behavior that might be induced by the chosen tasks. Because of this paradigm, there also is a gap in task complexity between the tasks typically used in cognitive science research and artificial intelligence studies employing planning algorithms. While the tasks from cognitive science research are mostly kept simple to make them easy to model and study, tasks from machine learning studies often are more complex for demonstrating the superb performance of new models (van Opheusden and Ma, 2019).

One way of easily providing tasks with different complexities is video games. A great variance of different tasks can easily be included within a single game that might resemble difficulties similar to the challenges of normal everyday tasks. Of course, there might (and probably will) still be differences between solving tasks naturally in a physical environment and solving tasks on a computer screen, but still, video games offer a great opportunity for studying human behavior in environments that are quite natural without the need for great expenses.

In general, video games are extensively used for studying human behavior and benchmarking performances of artificial intelligence in various domains. Zhang et al. (2019) provide a huge data set of different Atari games played by humans along with the respective executed eye movements for decision-making research. Wang et al. (2020) benchmarked a multi-agent collaboration model in an environment inspired by a popular video game and Pohlen et al. (2018) evaluated the influence of planning horizon size in different learning algorithms on their performance in different Atari games. It has also been found that the sequential decision problems within certain games are most likely quite similar in their properties, so strategies should transfer well between games and thus might also transfer well into the real, physical world (Şimşek et al., 2016).

Also, the experience in certain tasks or more specifically in video games (of a certain type) are easy to assess, since participants of a study quite well estimate how much time they spend regularly or have already spent in their lifetime on video games, in contrast to rather unclear and difficult to quantify questions like "How good are you in avoiding obstacles quickly". This also offers the opportunity to study behavioral differences between experts and novices in (video) games. Playing video games on a regular basis is a good training for reducing reaction times (Dye et al., 2009). Green and Bavelier (2003) have shown experience video game players also have bigger visual attention capacities and recover fast from attentional blinks (Shapiro et al., 1997), which possibly allows them to consider more information for planning.

As already mentioned in section 2.2, vision and planning are closely related, which suggests that differences in the active vision of trained video game players to novel video game players also mean differences in planning strategies and the according information sampling of the environment. For example, Ma et al. (2021) found

that experts in board games have a deeper planning horizon, and supposedly perform better because they are able to plan further ahead.

Additionally, the differences between expert and novice video game players open up the question on how well people can be trained on better planning strategies. For instance, Callaway et al. (2022a) designed an artificial intelligence to improve the planning processes of people. Similar approaches could eventually be used to train people on better planning strategies on the long-term and not only within a specific task.

Relevance of This Work

This work offers perspectives and advances for several domains, predominantly in the development of artificial intelligence and human-computer interaction. Some of these perspectives are presented here.

- **Prediction of Human Actions** - Since the planning horizon is inseparably connected with the next actions people might take, a better understanding of it also serves as a basis for predicting how people act overall. This is a sensitive topic, since it can be used for controlling and sabotaging people as well as benefiting their everyday life to make it more safe and comfortable. We will not discuss this in detail here, but the possibilities are plenty and also partly covered by the following bullet points.
- **Improvement of Human Computer Interaction** - By understanding the inner workings of humans and how they plan their actions, systems interacting with humans can be laid out in a way such that they do not interfere with a human's plan and join collaborative tasks in the most pleasant way for the human user.
- **Recognizing Safety Concerns** - Human behavior is not optimal in many regards, which also applies to planning processes (Nelson et al., 2018). For example, Carton et al. (2016) have shown that the human planning horizon is shorter when people need to re-plan their trajectory in order to avoid collisions with objects. This can pose risks to other people in everyday situations, for example when drivers of vehicles need to adjust their trajectory but cannot incorporate that the adapted trajectory might be dangerous for other people, because their planning horizon was too short. Proper understanding of the human planning processes can help addressing these safety concerns.
- **More Human-Like AI** - Advances in the understanding of human planning strategies can serve as a inspiration for the design for more human-like or even better artificial intelligence. These models can then be used for simulating human behavior in situations that might be un-testable otherwise. Also, we might be able to design more efficient planning algorithms if we understand how the human mind plans efficiently without exceeding its computational limits.

3. Experiment design

This chapter provides an overview over the environment (section 3.1) and its basic dynamics, as well as the hardware (section 3.2) used in the experiment. Section 3.3 then explains the basic design in terms of the procedure all subjects went through.

3.1. Environment

Our environment is greatly inspired by the video game *Frogger*, released by the *Konami Corporation* in 1981. We programmed our own, slightly modified version of it to ensure it fits our needs and to integrate the eye tracker appropriately.

This game is a good fit for our experiment, since it is simple to explain, such that even inexperienced subjects can quickly understand and learn it, while still comprising two fundamental tasks within a path-finding setting. These two tasks are a obstacle avoidance task and a slightly modified version of it, which will be explained in more detail in the upcoming sections. As explained in section 2.4, video games are a great way to examine human behavior in more naturalistic settings, while still having a reduced and controlled environment for certain tasks only.

3.1.1. Basic structure

An example level from the game in its initial state is shown in figure 3.1. The level consists of a 15 rows \times 20 columns grid. The player is represented by a frog, that needs to get to the finish lane at the top of the board within a certain time limit, while staying alive and inside the level's boundaries, after starting from the center of the first lane on the bottom of the screen.

The rows represent different lanes which the player has to cross. Namely, these are one *starting lane* at the bottom of the screen, followed by six street lanes forming the *street section*. Within this section, the player has to avoid colliding with randomly spawning, horizontally moving cars. After this, one *center* or *middle lane* similar to the starting lane follows, which does not pose any challenges on the player. The second section begins after that and again consists of six lanes. These six lanes represent the *river section*, in which the player must not touch the water by always staying on top of randomly spawning, horizontally moving water objects, represented by planks and lily pads, while also avoiding to be carried out of the screen by these water objects. There is no functional difference between planks and lily pads per se, since they behave completely the same. The difference is only the texture of their sprite and thus purely cosmetic to provide a better user experience. Water objects with a width $w = 1$ are represented by a lily pad while water objects with a width $w > 1$ are represented by planks. The top row is the *finish lane* and contains a target represented by a star, which the player needs to reach to successfully complete the level. The star in the finish lane is either placed on the left, center or right of the finish lane.

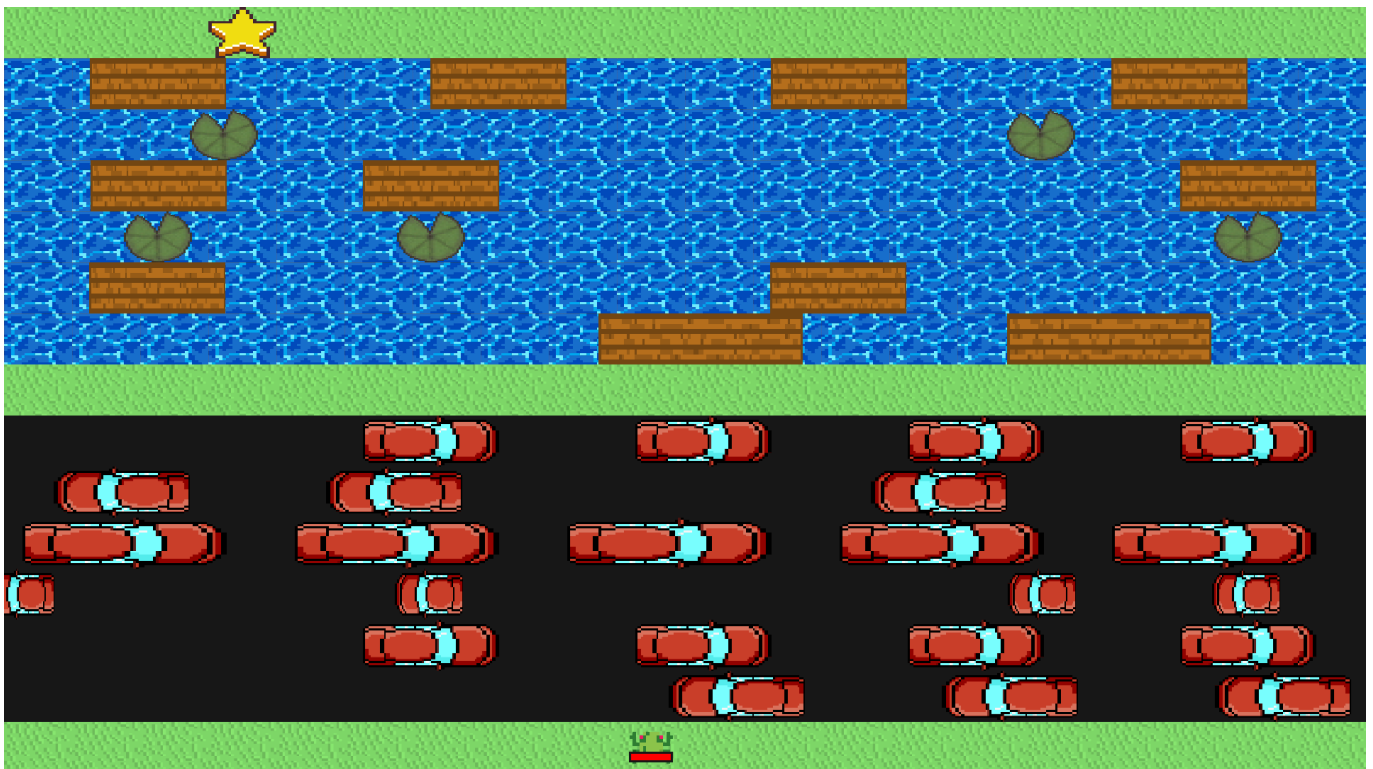


Figure 3.1.: Example level from the experiment. The game board consists of a 15 rows \times 20 columns grid. The board is divided into two sections, a street and a river section, which the player, represented by a frog, needs to cross successfully to win the game. A trial is completed successfully, once the player reaches the target position in the top lane, represented by a star. The remaining time for this trial is shown by a decreasing red bar below the avatar / the frog.

The two sections, river and street, pose two different tasks. The first task in the street section is a basic obstacle avoidance task, since the player must not run into the moving cars. The second task in the river section is a modified version of this, since the player must also avoid running into the water, but this time the player itself is moving even without taking any actions. Also, not taking an action can lead to failing the trial in the street section, while not taking an action does not pose a threat until the player is close to being carried out of the screen.

The player is represented by a frog and has attached a red bar decreasing in size from right to left, proportionally to the time left for completing the level. The time limit for completing the level is 75 seconds.

The player can move the frog in any vertical and horizontal direction by one field by pressing the arrow keys on the keyboard. The frog always only moves one field at a time. After each key press, there is a delay of 350 ms until the frog can be moved again.

The dynamics of each lane containing moving objects (cars in the street section, planks and lily pads in the river section) are determined by certain parameters. The lanes alternately move in different directions. River lanes and street lanes are defined by the object width w , the distance between the objects d , the velocity v and the spawn probability θ . All objects within one lane share the same parameters.

The parameters w and d are given in fields of the game board and define possible spawn locations for an object, as shown in figure 3.2. While the overall movements of all objects are strictly deterministic, the spawn probability θ employs a stochastic component. Whenever a spawn is possible, the probability θ defines a Bernoulli trial which decides whether an object really spawns or not. Namely, θ defines the probability of a successful spawn, which means that an object is actually spawned. The possible spawn locations are independent of whether the previous spawns were successful or not. The distance between two objects is always equivalent to $d + i \cdot (w + d)$ fields where i is the number of skipped spawns since the last successfully spawned object.



Figure 3.2.: Structure of a street lane in terms of object width w , object distance d , lane velocity v and spawn probability θ . The distance between two objects is always equivalent to $d + i \cdot (w + d)$ fields where i denotes the number of non-spawned objects between the current and the previous object. This also applies to river lanes in the same manner.

We defined a generative model for the generation of levels of different difficulties. Overall, there are three difficulties: *easy*, *medium* and *hard*. The generative model defines various Multinoulli distributions over all possible values for each street and river lane, depending on the level's difficulty, which is passed as a parameter into the generative model. This is shown in table 3.1.

The possible values and their respective probabilities were manually tuned using obvious relations between the meaning of parameters and how they influence the difficulty of a level. For example, faster moving cars or smaller lily pads make the level increase a level's difficulty. In principle, the influence of the parameters on the level's difficulty are reciprocal for the street and river lanes. While, for instance, bigger objects in the street section increase the difficulty since it is more difficult to avoid the cars, bigger objects in the river section lead to bigger planks / lily pads, making it easier for the player to get on top of them. Additionally, the distributions of the generative model were validated by playing several levels with different parameters. The generative model also generates the position of the star in the finish lane, by drawing from a uniform distribution whether it is placed on the left, center or right of the finish lane.

v	p_{easy}	p_{medium}	p_{hard}	θ	p_{easy}	p_{medium}	p_{hard}
36px/s	0.20	0.20	0.00	cars water objects	0.60	0.65	0.75
54px/s	0.80	0.75	0.80		0.80	0.70	0.65
72px/s	0.00	0.20	0.20				
w_{car}				$w_{\text{water object}}$			
1	0.60	0.35	0.10	1	0.00	0.15	0.20
2	0.30	0.50	0.65	2	0.30	0.40	0.65
3	0.10	0.15	0.25	3	0.60	0.40	0.15
4	0.00	0.00	0.00	4	0.10	0.05	0.00
d_{car}				$d_{\text{water object}}$			
-1	0.15	0.00	0.00	-1	0.00	0.00	0.00
1	0.00	0.05	0.10	1	0.15	0.10	0.00
2	0.40	0.65	0.80	2	0.65	0.50	0.15
3	0.45	0.30	0.10	3	0.20	0.40	0.85

Table 3.1.: Probabilities for drawing a certain parameter value for a specific row. The columns p_{easy} , p_{medium} and p_{hard} denote the probabilities for the different values shown in their left-hand column for different difficulties. Values for d and w are given in fields Please note that a distance of $d = -1$ corresponds to an empty row without any objects.

The direction of the lanes alternates left and right, as already mentioned above. Whether the first moving lane starts moving towards the left or the right is decided with a Bernoulli trial with $p = 0.5$ where the possible outcomes are the two directions left and right.

The goal of the game is to reach the star in the finish lane at the top of the screen without colliding with any of the cars or falling into the water. In detail, this means that the frog must not touch a car to avoid collisions in the street section and the center of the frog must always be on top of a plank or lily pad in the river section, to avoid falling into the water.

The star in the finish lane can only be entered from below, such that the player needs to find the right timing to get on a plank or lily pad, in order to be able to step on the star from below. The player cannot enter the finish lane anywhere on the screen and then walk to the star horizontally. By this, we preserve the need for more careful planning.

Each level can end with one of three possible outcomes. (1) The player successfully completes (wins) the game by reaching the star in the finish lane without touching any car, falling into the river or the time limit running out. (2) The time limit runs out, although the player did not touch any cars and did not fall into the water. (3) The player touches a car, falls into the water before getting to the finish lane or is carried out of the level while being on a water object. In this case, there will be time left, but this does not matter.

$$s = \begin{cases} 170 + (t_{\text{limit}} - t_{\text{game}}) & \text{(1) level completed successfully} \\ 5 \cdot l_{\text{max}} & \text{(2) time ran out} \\ -100 & \text{(3) level failed} \end{cases} \quad (3.1)$$

As drive for the players to perform well, we assign a score s for each completed level according to a certain

scheme. That is, for a successfully completed level (1) the player is awarded 170 bonus points plus the remaining time $t_{limit} - t_{game}$ as difference of the time limit and the current level time in seconds. If the time runs out, (2) the player is only awarded with the number of the highest visited lane l_{max} times five points. If the player touches a car, falls into the water or is carried outside of the screen, (3) the player receives a penalty of -100 points flat. This is shown in equation 3.1. If the achieved score is positive, it is also multiplied with a factor β depending on the level's difficulty, that is $\times 1$ for an easy, $\times 2$ for a medium and $\times 3$ for a hard level, forming the final score s_{final} , shown in equation 3.2.

$$s_{final}(s) = \begin{cases} \beta \cdot s & \text{if } s > 0 \\ s & \text{otherwise} \end{cases} \quad \text{with } \beta = \begin{cases} 1 & \text{easy level} \\ 2 & \text{medium level} \\ 3 & \text{hard level} \end{cases} \quad (3.2)$$

3.1.2. Differences to vanilla Frogger

Our version of the game has some minor differences to the original *Frogger* by *Konami*, which we will cover here.

The overall grid is smaller, since the original frogger has a 13 rows \times 14 columns grid. By choosing a slightly bigger grid, we provide more possibility for planning while still staying close to the original setting. The bigger grid is also slightly more difficult and thus needs more time to be completed. Therefore, we increased the time limit from 60 seconds in the original game to 75 seconds in our version.

Also, in the original game the player needs to fill all of five possible target positions in the finish lane in five trials of the same level by entering them, where already filled positions cannot be entered again. In our case, there is always only one clear target in the finish lane that must be reached. To cover some of the variability in the target position from the original game, our level's targets are randomly placed on the left, center or right of the finish lane during the generation of each level. If the player fails to fulfill this goal in the original game, he can try again until he lost a total of three lives. In our implementation, the player only has one try per level. Regardless of whether they fail or succeed, after that one try the player is presented with the next level.

In the original Frogger, there are also varying distances d between objects within a lane. Our approach keeps the same distance d between all objects within a lane, thus following a more consistent pattern and being more predictable, which should be easier to leverage for planning.

While the player can rely on the objects not randomly vanishing in our version of the game, the original game featured randomly dis- and reappearing water objects, again making the future course of the game more unpredictable in each time step. It also features collectables which provide bonus points when collected and carried to the finish lane, but we wanted to keep the game small and simple to only account for two tasks besides the general path-finding towards the target. These two tasks are a classical obstacle avoidance task with moving obstacles in the street section of the game and a modified version of that in the river section. In the river section, the task is modified since the obstacles are not to be avoided but the only places where the player is allowed to go. Also, the players themselves are also in constant motion as soon as they leave the middle lane and move onto the water objects.

3.2. Eye tracker and hardware

The experiment was conducted in a well-lit room on a monitor with a screen resolution of 2560×1440 px, dimensions of 595×335 mm and a refresh rate of 60 Hz. The participants were seated and fixated in a chin rest in front of the monitor, with a total distance of approximately 102 cm to the center of the screen.

The eye movements of the participants were tracked using an EyeLink 1000 Plus eye tracker with 60 Hz sample rate and nine point calibration.

3.3. Experiment design

For the experiment, the participants were seated in front of the monitor. Following that, informed consent was obtained along with some basic demographic information (age, gender and experience with video games) and the eye tracker was calibrated. Then, the participants were made familiar with the game's rules and controls as well as the procedure explained in the following.

The overall procedure consisted of five training trials to get used to the control dynamics of the game, followed by 60 trials for the actual experiment. We generated a total of 60 levels according to the generative model defined in table 3.1, of which 20 levels were of difficulty easy, medium and hard each, and which were all solved by each subject. The levels were presented in random order. For the training phase, three easy and two medium difficulty levels were randomly chosen from the totality of all 40 easy and medium levels and presented in random order.

After the first and each five levels, the player was presented with their current score in a high score, to intensify their intrinsic motivation for performing well in each level. Various studies show that the intrinsic motivation is actually promoted by the presentation of the player's current performance in a high score (Halan et al., 2010; Landers et al., 2017). There are also studies showing that high score leaderboards do not necessarily have a positive effect on the performance of the players, but they also could not find any negative impacts on the performance (Pedersen et al., 2017).

Before any participants completed the experiment, the high score was filled with random entries that simulate a single player's total score over the course of the experiment. Each fake level score was calculated by uniformly drawing whether the game was won, lost or the time ran out and assigning scores for that as defined in section 3.1.1.

At the end of the experiment, all subjects were asked to provide some information on the strategies they used and were given the opportunity to give some feedback on the overall experiment and game design.

3.3.1. Participants

Fourteen (six females, eight males) undergraduate or graduate students recruited at the Technical University of Darmstadt who received course credit participated in the experiment. All experimental procedures were carried out in accordance with the guidelines of the German Psychological Society and approved by the ethics committee of the Technical University of Darmstadt. Informed consent was obtained from all participants prior to carrying out the experiment.

4. Analysis

This chapter provides an overview over the methods we used for analysis, that are more complex and might need more introduction than a short explanation within the results section. Section 4.1 gives a short introduction to the types and characteristics of various eye movements, as well as to the algorithm we use for detecting said events in the samples collected by the eye tracker. In section 4.2, we introduce the *Mean Fixation Distance (MFD)* as one of the main quantities used for our further analyses.

4.1. Detection of eye movement events

This section shortly presents the most important types of eye movements in section 4.1.1, followed by an explanation on how we classify our collected eye movement samples as such (section 4.1.2). Sections 4.1.3 and 4.1.4 describe the process of detecting blinks and measuring the pupil size of the subjects, respectively.

4.1.1. Types of eye movements

The most important types of eye movements for our analysis are *saccades* and *fixations*. Usually, people direct their attention towards a new area in a visual scene by moving their eyes with a rapid ballistic movement. This is a so-called saccade, during which the actual visual nerve to the brain is cut-off, most likely due to avoid motion blur caused by the movement. The average duration of a saccade lies within 20 to 40 ms but is also linearly dependent on the amplitude, i.e. the distance covered, of the movement. Because of their ballistic nature, saccades can be conveniently detected in eye tracking data by their high velocities and accelerations. After the gaze of a person was shifted by executing a saccade movement, they probably want to gather visual information about the new location. To do so, they use so-called *fixations* between the saccades, in which the subject's eyes are aligned with and focused on the target. Normally, fixations last about 50 to 600ms, but can theoretically also last arbitrarily long. During a fixation, the eyes do not perform any movements besides some noise and oscillations in the control of the movements, called *drift* and *tremor*, and intermittent extraordinarily short saccades, called *microsaccades*. (Land and Tatler, 2009; Rayner, 2009; Carpenter, 1988; Findlay et al., 2003)

4.1.2. Algorithmic implementation

Since the eye tracker only provided us with a sequence of samples containing the current gaze position on the screen, we needed to employ methods for detecting and extracting saccades and fixations from such sequences.

To ensure a well-tested and reliable classification of eye movement events, and to avoid the need to implement this ourselves, we use the *REMoDNaV* algorithm by Dar et al. (2021), which is an adaptation of the eye movement event detection algorithm by Nyström and Holmqvist (2010). Another advantage of this algorithm is that it does not only allow for classifying fixations or saccades, but also more complex types of eye movements, like smooth pursuit, where the individual slowly follows a stimulus with their eye movements or so-called *glissades*, which are post-saccadic oscillations in the eye movements. This gave us the possibility to extend our analysis in more complex ways, even though we did not need that for the main findings of this work. Furthermore, the algorithm by Dar et al. (2021) also performs well for dynamic scene perception and not only static scene perception as many other eye movement event detection tools do. This suits our needs well, since a video game clearly provides dynamic scenes that need to be processed by the player.

Event detection algorithms can be classified into *dispersion-based* and *velocity-based* algorithms (Dalveren and Cagiltay, 2019). Dispersion-based algorithms are based on the durations the eye tracker samples covered certain regions (e.g. Hessels et al. (2018)), while velocity-based algorithms rely on the calculation of the inter-sample velocities and accelerations (e.g. van Renswoude et al. (2018)). They then employ simple thresholding principles to classify the events. For instance, high velocities and acceleration indicate that the measured samples were part of a saccade. The inter-sample velocity v_i of two samples s_i and s_{i+1} is calculated as angular velocity in visual degrees from the euclidean distance between their positions p_i and p_{i+1} and the time delta Δt between them as shown in equation (4.1).

$$v_i = \frac{\|p_{i+1} - p_i\|_2}{\Delta t} = \frac{\|p_{i+1} - p_i\|_2}{t_{i+1} - t_i} \quad (4.1)$$

We can calculate the inter-sample accelerations analogously by simply performing the same calculation with the positions p substituted by the calculated velocities v . The values for thresholding velocities and accelerations usually have to be found and chosen empirically. Often, this needs manual tuning of the thresholds as 'rule-of-thumb', which is unscientific and can be error-prone. This is not the case in the algorithm by Nyström and Holmqvist (2010).

Difficulties in our use-case - The algorithms by Nyström and Holmqvist (2010) and Dar et al. (2021) are laid out to perform on high-frequency data that was collected with high sampling rates in magnitudes of > 1000 Hz. The filtering window for removing noise with the SG filter was chosen to cover a duration of about 20ms, which still contained a lot of samples in the original paper, because of the overall high sampling rate. Since we used a sampling rate of only 60 Hz in our eyetracker, the filtering window contains only two to three samples. This resulted in unusually high angular velocities and accelerations, which we were also warned about by the algorithm during the processing of the data. Still, we validated the results by also employing a naïve dispersion-based approach for detecting fixations and a simple velocity-based approach for detecting saccades. Since the results were in similar ranges, we accepted this loss in accuracy because of our low sampling rate in favor of an overall well-performant algorithm.

The main concept of the algorithm by Nyström and Holmqvist (2010) belongs to the class of velocity-based event detection algorithms, where the velocities and accelerations are first smoothed by a Savitzky-Golay (SG) FIR smoothing filter of second order, in order to remove noise while preserving high-frequency details (Savitzky and Golay, 1964).

Then, velocity peaks are detected and a velocity threshold for classifying samples as saccades is found iteratively.

An initial threshold PT_1 is chosen arbitrarily within 100 to 300 $^\circ/s$. Then the mean μ_1 and standard deviation σ_1 of all values below this threshold are calculated and the next threshold PT_2 is chosen as $PT_2 = \mu_1 + 6\sigma_1$. This process is repeated until the threshold PT_i converges to a value, such that the difference to the last threshold PT_{i-1} is less than 1 $^\circ/s$. This avoids the need for manual tuning of the threshold.

These threshold are then used to classify saccades and glissades, where consecutive samples need to cover a duration of at least 10ms to be classified as a saccade. Saccades with a velocity bigger than 1,000 $^\circ/s$ or a acceleration over 1,000,000 $^\circ/s^2$ are assumed to be physiologically impossible and are not considered saccades, even when fulfilling all other criteria.

All samples that do not belong to a saccade or glissade, were not recognized as noise and cover a duration of at least 40ms, are classified as belonging to a fixation.

The REMoDNaV algorithm employs the same principles as the algorithm by Nyström and Holmqvist (2010), but alters the calculation of e.g. the thresholds for saccade-classification to make it statistically more robust. For a detailed explanation on how the original algorithm is extended, please consult the REMoDNaV paper by Dar et al. (2021).

In our analyses, we mostly use fixations since they represent the areas of interest most clearly and indicate how far the planning horizon of the subjects extended. The REMoDNaV algorithm outputs a start and end position for each fixation, which usually do not differ by a lot, since the eyes do not move substantially during fixations; still, we used the middle position between start and end position of each fixation as the fixated location. We mainly focus on the amplitude of the fixations, usually taken relative to the player's position when the fixation was initiated, as externalization of the internal planning horizon.

4.1.3. Blink detection

Besides the distinction between fixations and saccades, there are also phases during human vision, where the visual nerve is cut off and the individual is not able to see anything: when humans are blinking (Volkman et al., 1980; Bristow et al., 2005). These blinks typically last about 50 to 350 ms (Volkman, 1986; Fatt and Weissman, 2013; Schiffman, 1990). Whenever the eye tracker could not detect the subject's pupil within this range of 50 to 350ms, we classified this duration as a blink.

The blink rate of individuals is generally lower during more exhausting cognitive processes Holland and Tarlow (1972); Peng et al. (2006). For instance, previous works see blinking as a mechanism which adapts in a way such that it does not interfere with cognitive processing (Holland and Tarlow, 1975). In terms of the task, that is currently executed by an individual, the blink rate has been found to correlate inversely with the difficulty of the task (Drew, 1951).

In our experiment, the subjects often have to pursue the movements of water or street objects in order to avoid a collision or touching the water. Poulton and Gregory (1952) have shown that the blink rates of individuals increase slowly before and after visual tracking of a certain objects, while being decreased during the tracking itself. Furthermore, blinks often seem to occur predominantly during sensory perception before executing cognitive operations and after these deliberations are finished, thereby framing such processes somehow (Siegle et al., 2008). Recent studies show that timing of blinks is also essential to minimize visual information loss, resulting in synchronized blinks in high-performing subjects (Nishizono et al., 2021).

We calculated the blink rates by counting the number of detected blinks and dividing by the total trial time. Analogously, we calculated blink rates specifically for the river and street section, by summing the time spent in each section and then dividing the number of blinks while the player was standing in one section by the

time spent in the respective section. If a player did not get to the river section or immediately died when leaving the start lane and thereby entering the street section, we did not calculate a blink rate for this section.

4.1.4. Measuring pupil size

The eye tracker we used (cf. section 3.2) also allowed for measuring the currently visible area of the pupil as measure for the current pupil size. We can use the pupil size as an indicator for different cognitive processes. Recent works have shown, that the pupillary response of an individual consists of higher pupil dilations when performing more cognitively demanding tasks (van der Wel and van Steenbergen, 2018; Beatty, 1982; Hess and Polt, 1964; Peng et al., 2006). As introduced in section 4.1.3, increased blink rates also correlate with higher cognitive loads. There is also work that shows that pupil dilation often happens after blinks during cognitive processes, thereby also creating a relation between higher blink rates and increased pupil sizes (Fukuda et al., 2005). Lee and Lee (2017) have found that the pupil dilates with increasing difficulty, specifically studied during playing video games.

Since the pupil sizes of people are inherently different for different individuals, we performed a z-standardization using the data for the pupil area over all completed trials per subject (Attard-Johnson et al., 2019; Medathati et al., 2020).

For inter-subjects comparisons, which we used for example in section 5.3, the z-score does not suffice, as we perform the z-standardization per subject. By definition, the mean of each subject's z-score is at 0. In this case, we simply normalized the pupil sizes of all subjects into the range $[0, 1]$, to enable us to compare the pupil sizes also between subjects.

Samples where the eye, for instance, is already half-closed but the eye tracker could still track the gaze, are not thrown off. The blink detection only classified samples as blinks, where the pupil could no longer be detected at all. Thereby, our data for the pupil size gets distorted, since the pupil size is reduced shortly before and shortly after a blink because of the partially-closed eyelids. To keep the data for analyzing the pupil size clean, we dropped the two samples before and after each blink, which corresponds to a time frame of approximately 33 ms before and after the pupil was completely covered by the eyelid during a blink.

Some pupil sizes exhibited a z-score of more than 20. Since pupils will most likely not suddenly expand more than 20 times the standard deviations away from the mean, we removed these samples as noise in the eye tracker measurements.

4.2. Accounting for different fixation durations

When reasoning about the planning horizon, we want to infer the current planning horizon by observing the recent eye movements, as explained in section 2.2. Thereby, we want to account for the distance, i.e. how far away from the agent the points of fixation were, as well as the duration of the fixations. To do so, we cannot naïvely take the sum of all fixation amplitudes and divide by the total number of fixations, since this would not take the fixation durations into account. To handle this, we introduced a new metric, which we simply call the *Mean Fixation Distance (MFD)*.

Assuming that the player currently is at field $(x_{\text{Player}}, y_{\text{Player}})$ and performs n different fixations \mathbf{F} on the respective fields (x_i, y_i) with a duration of δ_i each, where $i = 1, \dots, n$. We can now calculate the MFD as given in equation 4.2, by also making use of the Manhattan metric for calculating the distances between the player and the points of fixation.

$$\text{MFD}(x_{\text{Player}}, y_{\text{Player}}, \mathbf{F}) = \frac{1}{\sum_i \delta_i} \sum_{(x_i, y_i, \delta_i) \in \mathbf{F}} (|x_{\text{Player}} - x_i| + |y_{\text{Player}} - y_i|) \cdot \delta_i \quad (4.2)$$

Still, please note that this measure is ambiguous. Since we weight each fixation's amplitude with their respective duration, a MFD of 2 could come about by multiple fixations with a amplitude of 2 which lasted 1 second each. But it could also be the result of one fixation with an amplitude of 10 and a duration of 1 second and another fixation with amplitude 2 and a duration of 3 seconds.

5. Experiment results

This section is dedicated to the results from our experiment. That is, the performance of the subjects, how they moved across the levels overall and also the characteristics of their eye movements. Therefore, we use the methods introduced in section 4.

Section ?? shows how many trials were won, how the subjects scored and how long various trials lasted. We also visualize the movements of the players across the game board and how the subjects' fixations were overall distributed across the levels.

Another aspect of our analyses is the adaption of the planning horizon between different tasks on-the-fly, that is how the subject adapt their planning horizon when transitioning from the street section to the river section, as these two region supposedly pose similar but different tasks. Section 5.2 examines the mean fixation distances introduced in section 4.2 for the different regions, how the planning horizon extends spatially in the different sections and how physiological measures like the blink rate and pupil size correlate with the river and street sections.

The analyses for the river and street section are then extended and studied regarding their correlation with the subjects' performances in section 5.3. Again, we examine the MFD, blink rate and pupil size, but this time dependent on the score achieved by each subject.

5.1. Descriptive statistics

In order to get a better understanding of how people perform at all in the given tasks, it is favorable to take a look at and investigate the subjects' overall performance and behavior. This section is divided into three subsections, of which the first one is dedicated to the time needed to complete trials and how well the subjects performed or, namely, what scores they achieved (section 5.1.1). Section 5.1.2 presents how the players moved within the levels and the last section shows how their gaze was distributed while playing (section 5.1.3).

5.1.1. Performance

The overall performance of all subjects is defined by each subject's score and the count of the various trial outcomes - whether a trial was completed successfully by reaching the top row of the level, the time in a trial timed out or the trial was lost, as defined in section 3.1.1. If a trial was completed successfully, we can also analyze this by taking into account the time needed for completing the trial.

The different trial outcomes per difficulty are listed in table 5.1 and shown in figure 5.1. For a detailed overview of the performance of each subject, please consult figure A.1 in the appendix. The overall trial outcomes show a decrease in win rate accompanied by an increase in loss rate for harder difficulties. This

shows that our choice of the generative models for different trial difficulties (cf. section 3.1.1) was appropriate. The rate of timeouts is quite low overall, making up only 6.81% of all trials. In the easiest setting, almost no trials resulted in a timeout. Still, there is a slight increase in the proportion of timed out trials for harder difficulties, suggesting that more difficult levels require more time to be solved, even if colliding with a street object or moving off a water object is evitable. This further supports our choice of the generative models for the different level difficulties.

The average ratio of successfully completed (or ‘won’) trials for each subject is about 59.03% (median of 59.17%, variance of 0.0068). The most trials were successfully completed by subject *G* with 75% won trials, while subject *J* performed worst by having completed only 45% without the time running out or losing the trial.

	Easy	Medium	Hard	
Won	26.81%	19.44%	12.78%	59.03%
Timeout	0.42%	2.92%	3.47%	6.81%
Lost	6.11%	10.97%	17.08%	34.16%
	33.34%	33.33%	33.33%	100.00%

Table 5.1.: Matrix of different trial outcomes by difficulty in percent over all $n = 720$ trials solved by all 12 participants. Bottom row and right column show the marginal proportions.

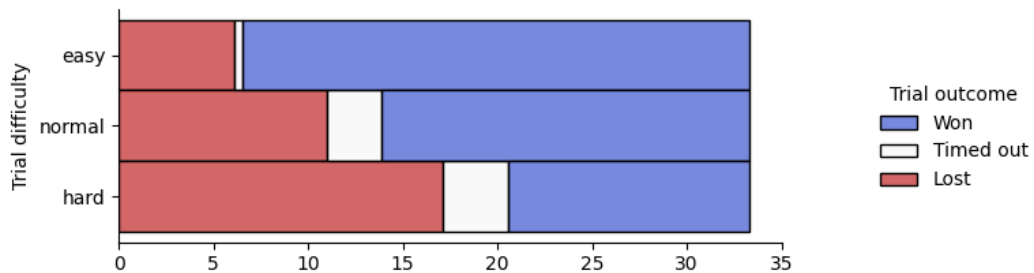


Figure 5.1.: Distribution of rial outcomes in percent over all $n = 720$ trials solved by all 12 participants.

The subjects’ performance is not only characterized by the number of won trials, but also by how well they performed in each trial. The won trials can also be distinguished in ‘better’ or ‘worse’ trials by how fast the players were able to complete the level. A histogram over the trial times for all successfully completed trials over all subjects is shown in figure 5.2. The average time a subject needed to complete a successful trial was 40.5394 seconds (median of 38.52 seconds , variance of 207.5762 seconds²) of the 75 seconds time limit of each level. The histogram shows a weak tendency for harder levels to need more time to be completed. The average trial times are shown in table 5.2.

Difficulty	Time Mean (Variance) [s]	Time Median [s]
Easy	36.2552 (202.4472)	35.346
Medium	42.6913 (190.7542)	41.833
Hard	46.2524 (169.6057)	44.106

Table 5.2.: Average trial times for all 425 successfully completed trials over all 12 subjects.

How fast the subjects were able to complete a level is also present within their level scores. Since the score for each won level was composed of a constant bonus for winning the level and higher bonus points the more

time of the level's time limit was remaining, the only possibility to achieve a higher score when winning a level was to complete it in less time. The detailed scoring criteria were introduced in section 3.1.1.

The average level score of each subject is shown in figure 5.3. The average score per level over all subjects was 291.0049 points (median of 304.5815 points, variance of 120241.2497 points²). While most subjects had an average level score around the range of 250 to 300 points, subjects *I* and *G* performed outstandingly well and subjects *J* and *L* performed worst, yet not as distinctly as subjects *I* and *G* performed well.

We can leverage this scoring information by combining it with other behavioral data and search for patterns in better or less successful participants. Thereby, we can make assumptions about which behaviors help to perform better or worse. Section 5.3 is dedicated to this purpose.

Uncertainty in level scores - Since the final state of each trial was not saved, we needed to calculate estimates for the scores of each level. Because we have sampled the complete state and action space with a 60 Hz sampling rate in accord with the eye tracker, we have access to the state less than 17ms before the trial finished and thus our estimates should be very accurate. Still, these estimates might be a little noisier than if we had access to the directly recorded final state and score of each level.

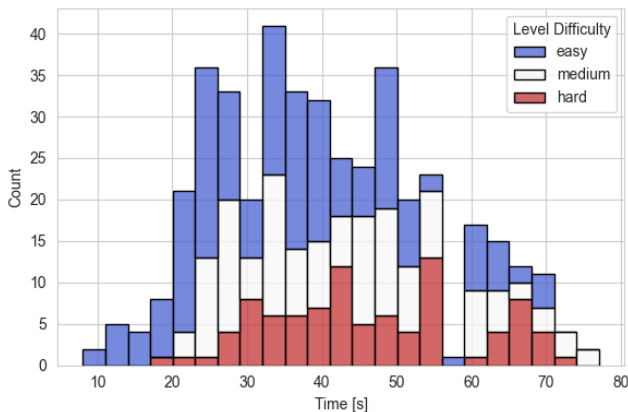


Figure 5.2.: Histogram over the trial times for all successfully completed trials in different difficulties / conditions.

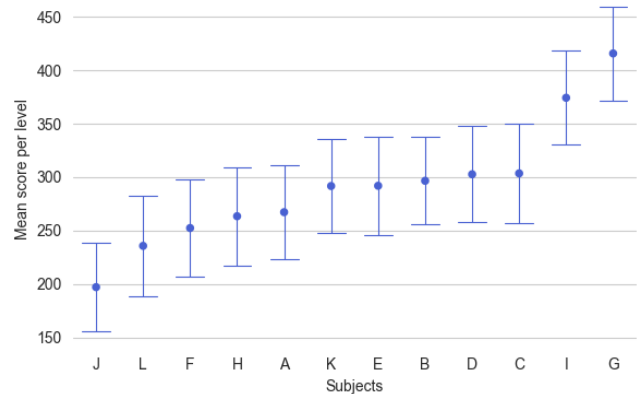


Figure 5.3.: Average trial score per level over all subjects, including all trials (i.e. won, timed out and lost trials). Error bar shown is the 95% confidence interval.

Besides the performance in the successfully completed trials, we also investigated in which lane the timed out and lost trials ended. This is shown as a histogram in figure 5.4. The histogram shows the lane the player was on immediately before they took the next action that led to them losing the trial or touching a car in the street section or the water in the river section. Because of this, there are also non-zero trial endings for the start lane (lane 1) and the middle lane (lane 8), where the subjects sometimes stepped off and lost immediately. The histogram shows that all timed out trials ended in the river section, meaning that all players either were able to successfully complete the street section or losing the trial before the time ran out. The most trials ran out of time when the player was on the last or second-to-last to row before the finish lane (lanes 13 and 14). This is in line with feedback from the subjects, which they gave at the end of the experiment. Most subjects said that they often reached the end of the river section but were not able to get to the target, because especially in the harder levels the water objects rarely spawned such that they offered a path directly to the target. In these cases, the subjects sometimes waited for the water object they were on to reach the target, whilst the time slowly ran out.

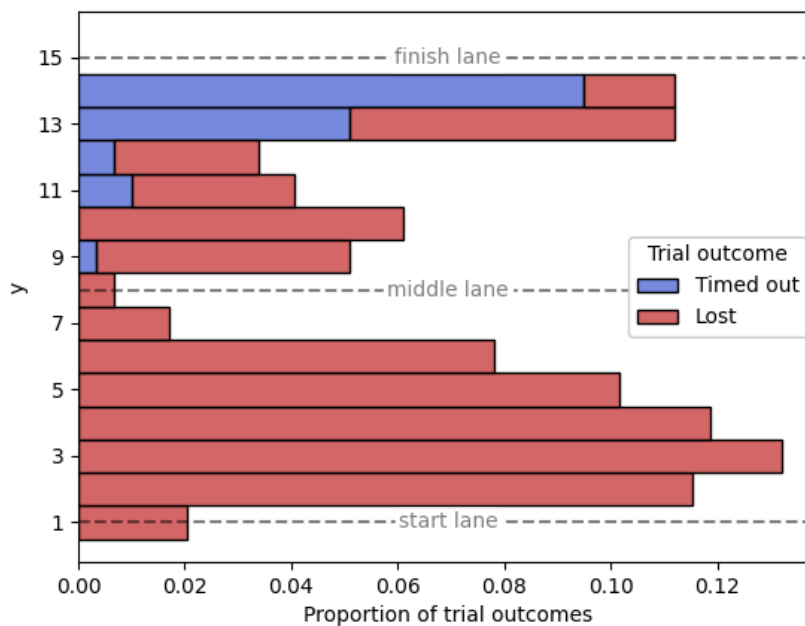


Figure 5.4.: Histogram showing the player position immediately before a trial was lost or when the trial's time limit was reached. Most trials ran out of time at the end of the river section, because people had to wait for a water object to provide access to the target.

5.1.2. Movements / Behavior

It is not only of interest how well the subjects performed, but also how they moved across the board, to investigate their overall movement patterns.

Figure 5.5 shows the distribution of time spent on each field of each level for all subjects. The starting position had to be set to zero in the heatmaps because it took on most of the time spent in the levels. This suggests that the subjects would still take some time at the start of each trial to plan the best way to get to the target. This seems to be similar in the middle lane (lane 8). Though the time spent there is not as high as in the start lane, the subjects spent some time in the middle lane, probably planning the next steps. On average, the subjects spent 7.1579% of the complete trial time on the middle lane (median of 4.6872%, variance of 0.007299). The time spent in the last lane of the street section, right before the middle lane, is almost none. This probably comes about because the players, once they reach the last lane of the street section, can immediately move on to the middle lane, where they are not exposed to any threats.

Besides, we can see a strong tendency to stay near the horizontal center of the level. This is beneficial in both the street and the river section, since staying near the center enables the subjects to leverage the most information in each lane on which street or water objects were already spawned and when they will enter the vicinity of the player before taking the next action. If the subjects moved near the left or right edges of the level, they could, for instance, not see when a car is spawning and would be ran over immediately. Still, the paths of the players spread a little to the left and right in the river section, since they need to move to a target on the left or right side of the screen in some trials (cf. section 3.1.1). Because this spread rather occurs in the river section, it is likely that the players recognized the middle lane (lane 8) as some kind of 'checkpoint',

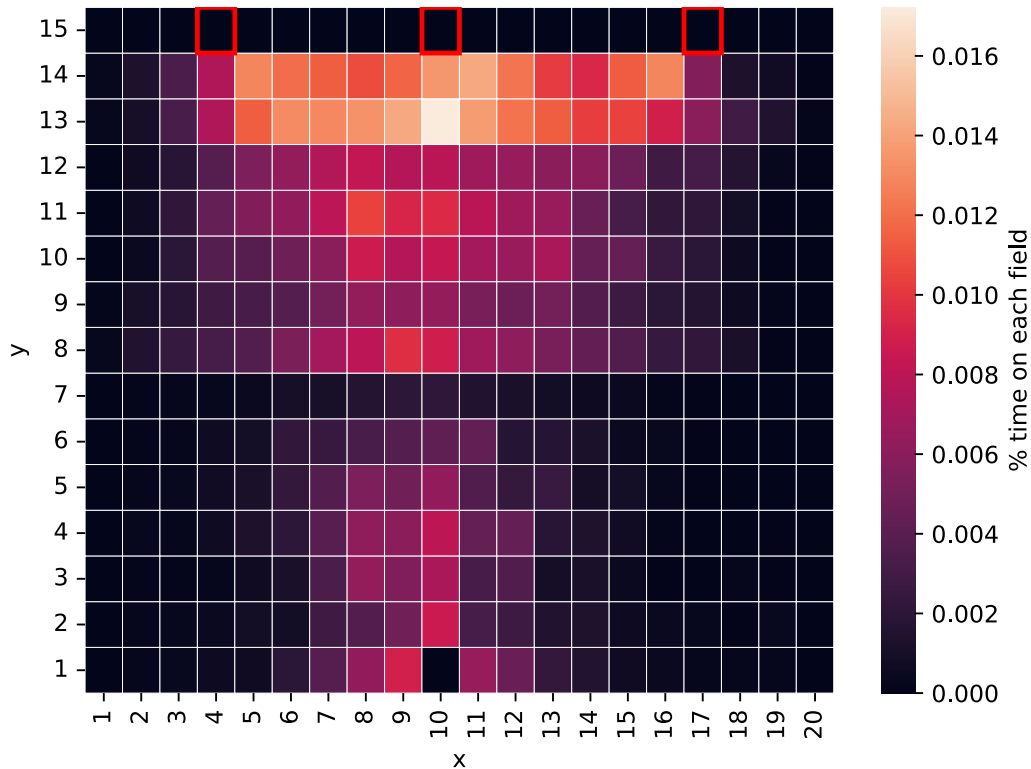


Figure 5.5.: Heatmap showing the time spent on each field as proportion of the total time over all trials of all subjects. The possible target positions are highlighted by red rectangles. The start position was set to zero for the sake of the color bar's scaling.

where they cannot die, early on and thus did not consider the target position while still being in the street section.

This is also apparent in figure 5.6, where the same heatmap from figure 5.5 is shown but for different target positions. In these heatmaps, the fields with the maximum time spent on are always near the target positions. Horizontally, we also see a tendency to spend more time near the position of the target in the middle lane here. This suggests that the subjects might take the target position into account now for their planning in the middle lane, in contrast to the start lane.

To further examine whether the subjects already considered the target position when entering the river section, we also extracted the x positions when entering the river section from the middle lane, shown in figure 5.7. While there is a small tendency to enter the river section slightly more near the target, i.e. to the left or the right of the center, the subjects usually entered the river section near the horizontal center of the level. Still, figure 5.6 shows that the subjects horizontally spent more time near the target positions, potentially externalizing their deliberations with their movements.

This could suggest that the subjects actually take into account the target position when entering the river section. Yet, it still might have been necessary to enter the river section at the center, even when the target was placed on the left or right, in order to get onto a water object which carries the players to the target position later on and to have the most information on where and which water objects have spawned. Thereby, more time is spent towards the target positions in the middle lane, although it is actually entered nearer to the center.

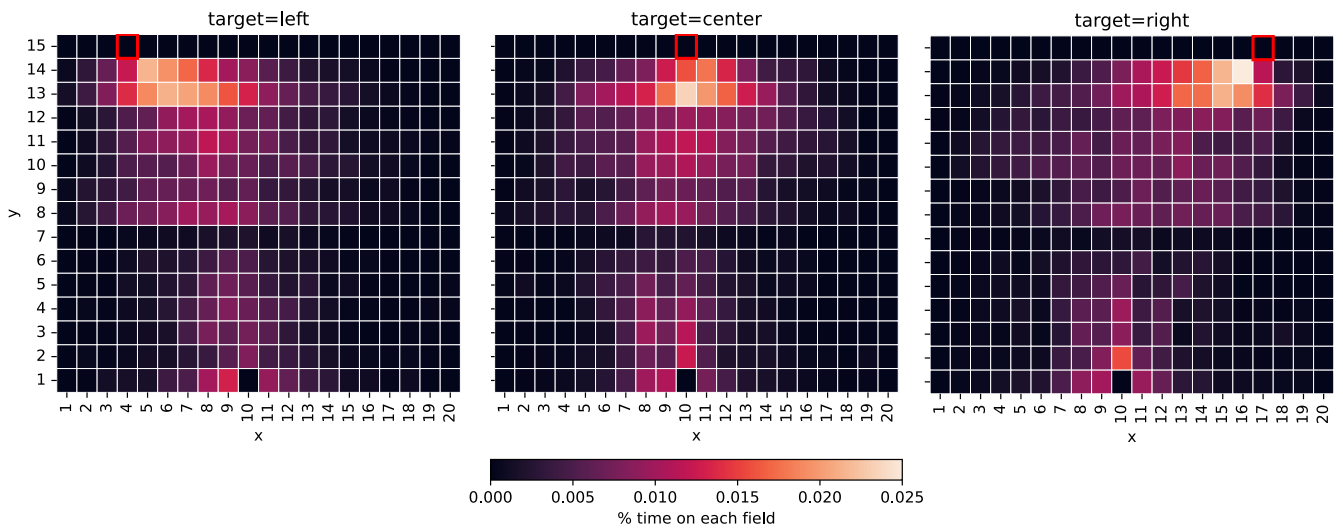


Figure 5.6.: Heatmap showing the time spent on each field as proportion of the total time over all trials of all subjects, separated by the target position of each target. Again, the respective target positions are highlighted by red rectangles and the start positions were set to zero for the sake of the color bar's scaling.

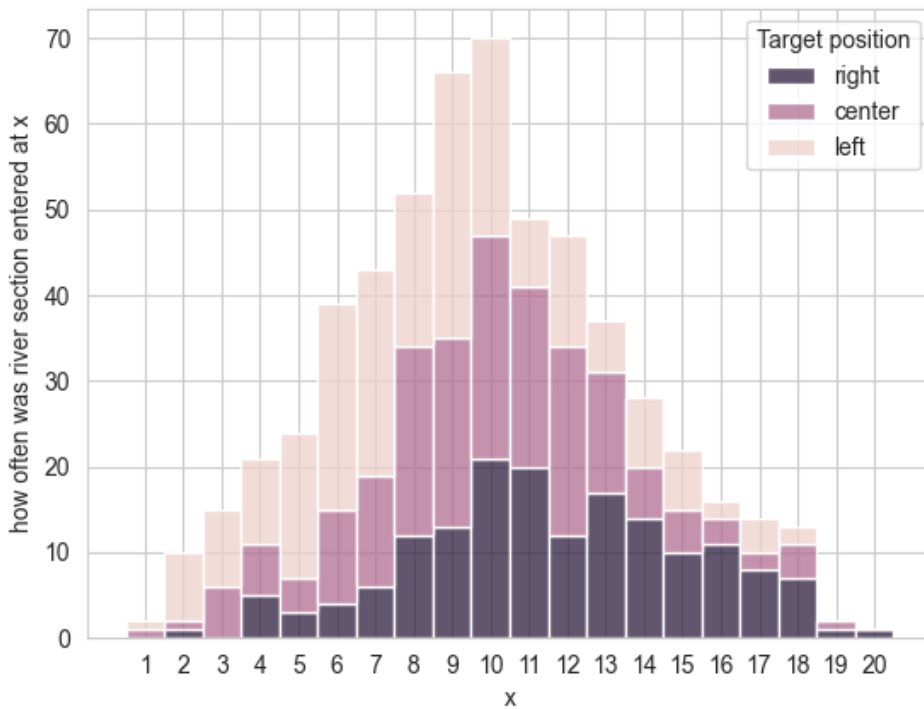


Figure 5.7.: Horizontal position of players when entering the river section from the middle lane, by target position of each trial. We see a weak tendency to enter the river section near the target's horizontal position.

5.1.3. Gaze

The main part of this work focuses on the movements of the subjects' eyes, as eye movements serve as some kind of externalization of the planning horizon by revealing how far ahead individuals looked in certain situations (cf. section 2.2).

The distribution of fixation duration over the whole level grid is shown in figure 5.8. We only considered successfully completed trials for this, since a lot of trials were already lost in the street section and thus induce more mass near the beginning of the street section in the fixation distribution. The heatmap shows that most of the fixations occur near the horizontal center of the street section and near the three different target positions in the river section.

From figure 5.5, we know that the subjects primarily crossed the street section in the horizontal center of the level. Therefore, it makes sense that a lot of the subjects' fixations land there.

The fixations near the target position in the river section also seem reasonable, since the players must of course look at the paths they need to take to get to their targets.

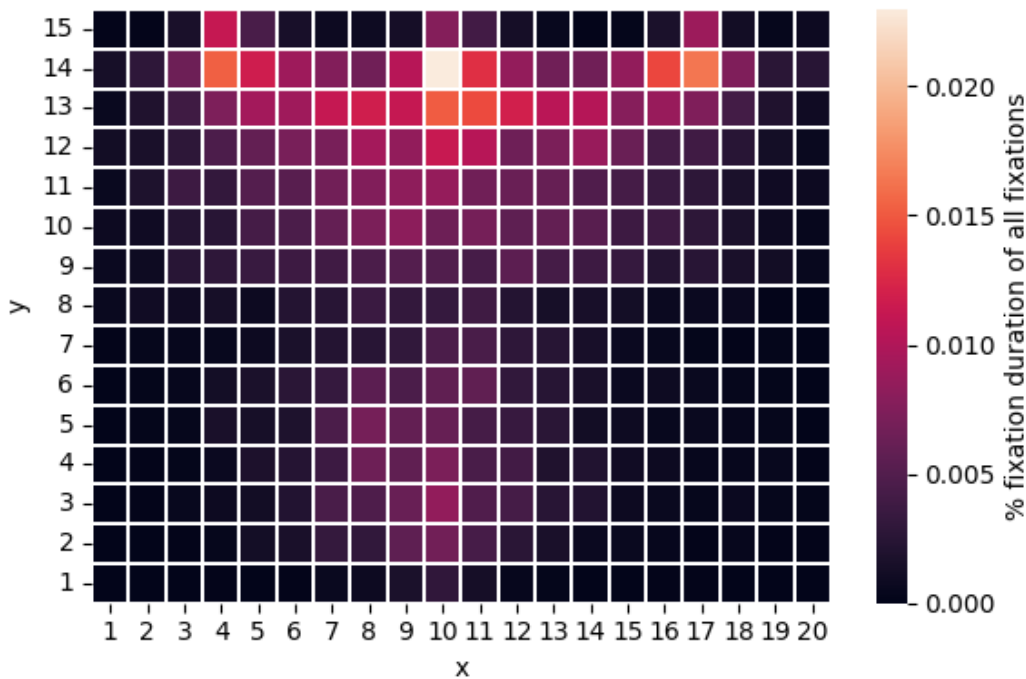


Figure 5.8.: Proportion of fixation duration on each field over all successfully completed trials of all subjects.

Still, figure 5.8 is not able to visualize the temporal characteristics, when people fixated where. This is covered in more depth in sections 5.2.1 and 5.3.1 by using the *Mean Fixation Distance (MFD)* introduced in section 4.2. For now, figure 5.9 visualizes the fixations on the target, per lane which the player was on during the fixation. We classified a fixation as *on target* if the fixation fell onto the target field or as *near target* if the fixation fell onto one of the fields vertically, horizontally or diagonally next to the target field. The figure shows that the subjects only started to really consider the target in the river section, especially towards the end of it. Almost no fixations on the target were made in the street section, except for a few target fixations while the players still were in the start lane.

This is also in line with the movements of the players in figures 5.5 and 5.6. The players mostly adjusted their

movement trajectories towards the target positions in the second half of the river section, where they also spent the most time while fixating the target or its immediate vicinity. Naturally, the players carry out more fixations on the target, when they are moving closer to it. Also, scanning the vicinity of the player when they are moving near the target contributes to more fixations on and near the target as well.

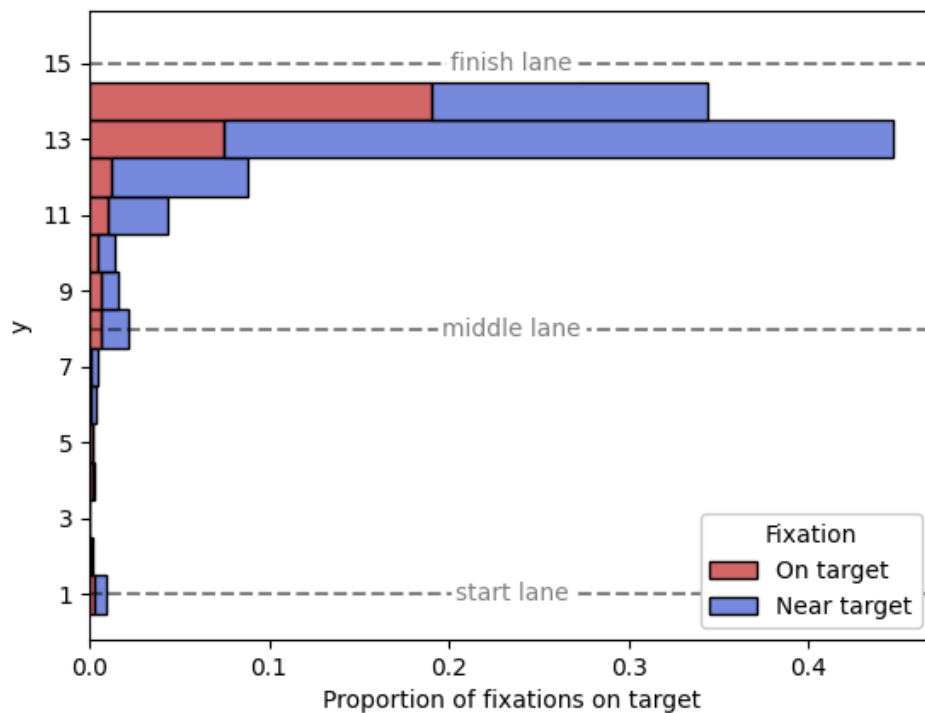


Figure 5.9.: Distribution of fixations on the target field or near the target field per lane the player was on during the fixation.

5.2. River versus street

It lies in the nature of the game Frogger, that the players face two similar, but different tasks in the river and street section.

In the street section, the players need to solve a classical obstacle avoidance task by crossing the street without colliding with any vehicles. In the river section, they need to avoid falling into the water. This is basically the same as in the street section, yet it differs in that the player itself is moving and the water objects, which provide safety, are also moving. In the street section, the safe fields were stationary and only the cars were moving. Since the safe fields in the river section only consist of the water objects, which only make up a small part of the area, especially in harder difficulties the river section is expected to need more careful and extensive planning.

This section investigates said differences between the two sections of the environment. Section 5.2.1 looks into the MFD (introduced in section 4.2) as a qualitative measure of the length of the planning horizon for the

different sections. Section 5.2.2 explores where the players' fixations and thus also their planning horizon is directed at, since the planning horizons can extend variably in different (spatial) dimensions. The last two sections examine blink rate (section 5.2.3) and pupil size (section 5.2.4) as physiological measures of cognitive load. We would expect the cognitive load to be higher in areas with higher MFDs, since the subjects need to consider more information when employing bigger planning horizons.

The collected data comprises 73,530 fixations of all 12 subjects, of which they executed 7,381 fixations in the start lane, 14,982 fixations in the street section, 6,011 fixations on the middle lane and 45,156 fixations while in the river section.

Though we expected to have less fixation data in the river section, since players often already died before reaching the river section, we have significantly more data there. This is probably due to the players spending more time in the river section than in the street section, also because of the players more often needing to wait some time for specific water objects to align such that they can cross the river, and thus offering more time for executing fixations.

5.2.1. Mean Fixation Distance (MFD)

The most obvious measure to obtain a good estimate of the planning horizon of people is by calculating the distances at which they fixated the screen. In a case like ours, where the subjects are viewing their avatar from a bird's-eye view, it is beneficial to investigate these fixation distances on the screen relative to the player's position during the fixation. For that, we introduced the so-called *Mean Fixation Distance (MFD)*, described in more detail in section 4.2.

We calculated the MFD with respect to each position of the environment grid. The means of all subjects' MFDs per position are shown as a heat map in figure 5.10. Qualitatively, the heat map suggests a distinctly higher MFD in the river section than in the street section. The mean MFD over all positions in the street and river lanes is visualized in a box plot in figure 5.11. The overall MFD seems to really be higher in the river section than in the street section, suggesting a bigger planning horizon for this section and task.

A Kolmogorov-Smirnov test indicates that the distributions of the MFD for the river and street area are significantly different ($D = 0.193$, $p=0.0$) and a two-samples t-test also shows that the mean MFD is significantly lower in the street area than in the river area ($t(60382) = 36.13$, $p < 10^{-283}$).

These results are in line with our expectations, since the river section poses a more difficult problem on how to cross the section, because of less space to move on in a more dynamical setting, thus requiring more planning and planning further ahead.

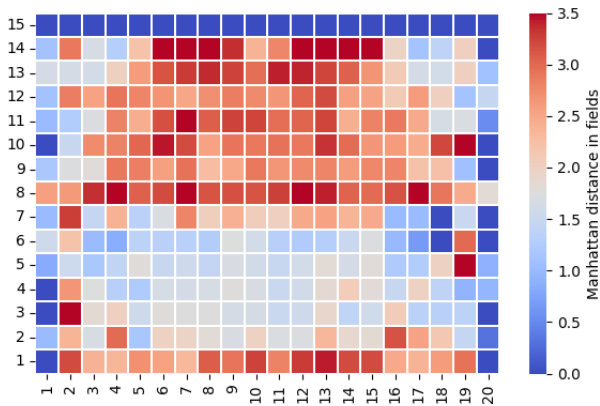


Figure 5.10.: MFD for all player positions over all subjects and levels in fields. Fields with missing data are filled with 0.

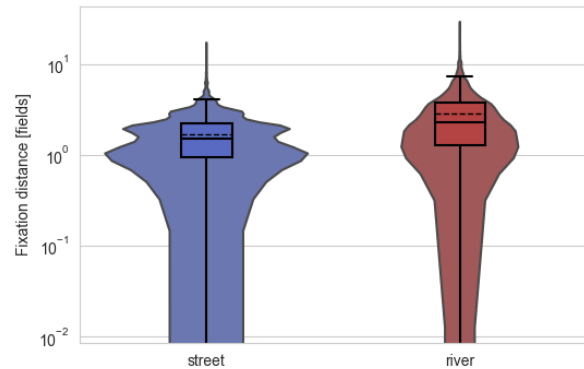


Figure 5.11.: Distribution of distances between a fixation and the player's current position weighted by each fixation's duration for the street and river section of each level. Please note the log scale of the y-axis.

5.2.2. Direction of fixations

We calculated the distances of the fixations relative to the current player position in x- and y-direction to obtain a more clear image of the directions the planning horizon is directed to. The characteristic values are shown in table 5.3 for the x- and y-direction.

x-direction [fields]	min	max	mean (var)	median
Total	-0.8601	1.1765	0.0042 (0.0018)	0.0000
River	-0.8601	1.0168	0.0043 (0.0016)	0.0000
Street	-0.5181	1.1765	0.0029 (0.0015)	0.0000
y-direction [fields]	min	max	mean (var)	median
Total	-2.1927	2.2930	0.0114 (0.0027)	0.0032
River	-2.1927	1.0194	0.0024 (0.0016)	0.000
Street	-0.0783	1.7818	0.0222 (0.0033)	0.0076

Table 5.3.: Characteristic statistics for fixations relative to the current player position in x- and y-direction in pixels.

The overall distribution of fixations relative to the player position during the fixation was computed using Gaussian kernel density estimation (KDE) and is shown in figure 5.12. Qualitatively, there are some characteristic properties to notice here. First, the fixations have their mode clearly in the area near the player. These fixations near the player would represent a short planning horizon, since they only consider the immediate vicinity of the player.

Moreover, there is a clear skew in the distribution for the street section, since there is more mass in positive y-direction. This makes sense, since the players' goal is to move forward and reach the finish lane at the top of the screen. Still, it is not that pronounced for the river section. While the street section seems to almost only fixate areas in front (positive y direction) of the player, the subjects also fixated more of the area behind (negative y direction) them in the river section. Besides that, the fixations are more spread in x-direction

for the river section, suggesting a change in the subjects' planning strategy. When moving towards the river section, it seems they consider even going back some lanes and thereby more possibilities to move forward, indicating a longer planning horizon in the river section.

Also, the spread of fixation direction in the river section seems to have a bias towards straight horizontal or vertical directions. Except for the peak to the top right, most density is not spreading diagonally but along the x- and y-axis.

Besides, there are two peaks at the top of the distributions, particularly visible in the distribution regarding the river section. The most distinct one is visible at $x = 0$ pointing in positive y direction. Another points towards north-east. These peaks most likely come about because of the different target positions in different levels. The target the player must reach is either located on the left, right or center of the top lane (cf. section 3.1.1). Thus, whenever a player looked at the target position to orient themselves, they looked into one of that directions. There is no peak towards north-west visible, but we attribute this to noise, since the generative model for the level generation used a uniform distribution over the possible target positions in each level and thus generated similar amounts of levels for each target position left, center and right.

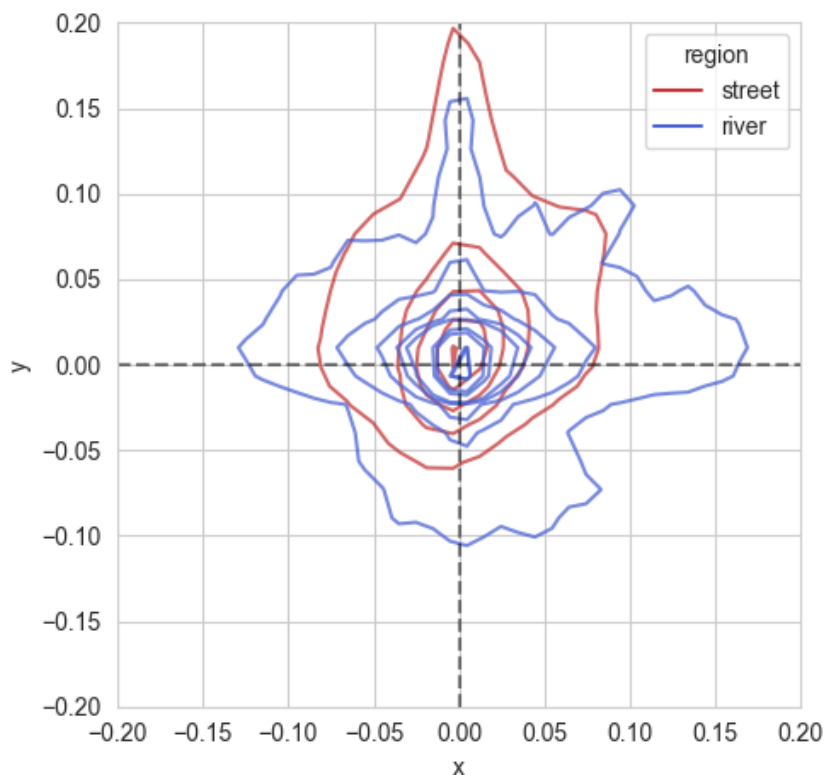


Figure 5.12.: Kernel Density Estimation Distribution with a Gaussian Kernel over all fixation positions relative to the player position during the fixation in fields for the street and river section. Marginals are shown as histogram ($n_{street} = 29265, n_{river} = 8856$).

To verify the aforementioned assumptions, we used t-tests. Regarding the fixations in y-direction relative to the current player position, a two samples t-test indicates that the difference in y-direction is significantly bigger in the street section than in the river section ($t(38119) = 36.3535, p < 10^{-284}$).

The fixations in x-direction relative to the current player position on the other hand seems to be rather spread out in the river section compared to the street section. A two samples t-test indicates that the difference in

x-direction is significantly bigger in the river section than in the street section ($t(38119) = 2.8000, p < 0.0026$). These results altogether suggest that there is a shift from a vertically oriented planning strategy to an rather all-around or at least more horizontally aware planning strategy when moving from the street section to the river section.

5.2.3. Blinking behavior

We classified missing eye tracker samples within a certain time frame as blinks (cf. section 4.1.3). Since the blink rate during tasks correlates with the (perceived) difficulty of the task by the subject, this can also unveil cognitive processes. For example, more demanding tasks exhibit larger cognitive load and thus we should observe a decrease in blink rate in these situations. For more details on this, consult section 4.1.3 again.

The blink frequency over all games for all subjects had a mean of 0.2311 blinks/s (median of 0.1583 blinks/s, variance of $0.041 \text{ (blinks/s)}^2$). For the street section, we observed a mean of 0.1175 blinks/s (median of 0.0 blinks/s, variance of $0.0392 \text{ (blinks/s)}^2$), while the mean in the river section was 0.2365 blinks/s (median of 0.1715 blinks/s, variance of $0.0501 \text{ (blinks/s)}^2$). A Kolmogorov-Smirnov test suggests a significant difference between the distributions of blink frequency in the street and the river section ($D = 0.4894, p < 10^{-67}$) and a two samples t-test shows that the blink rate is significantly higher in the river section than in the street section ($t(1258) = 9.9817, p < 10^{-22}$). These results are visualized in a box plot in figure 5.13.

We did not expect these results at all. The observed higher MFDs in the river section indicate longer planning horizons and thus should employ higher cognitive loads to handle more information and more extensive planning. On the other hands, previous blinking research has shown that blink rates decrease with more cognitively demanding tasks (cf. section 4.1.3).

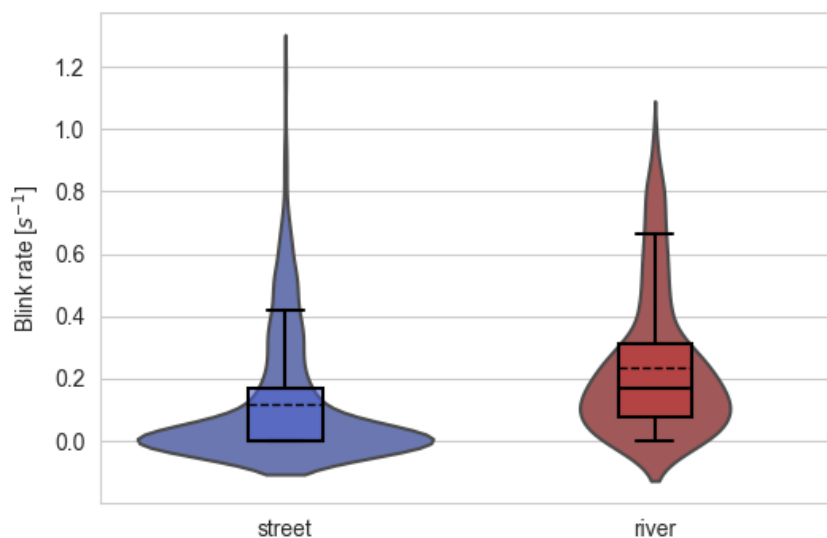


Figure 5.13.: Blink rates in blinks per second for all trials, separately for the river and street section ($n_{street} = 714, n_{river} = 546$).

5.2.4. Pupil size

We also took a look on the size of the subjects' pupils during the course of the game, since stronger pupil dilations can reflect higher cognitive engagement, for instance in tasks that are demanding more extensive planning. To account for the inter-individual difference in pupil sizes, we performed a z-standardization on the measured pupil sizes of all subjects and used these for analysis (cf. section 4.1.4).

By definition, the z-score of the pupil size for over all subjects had a mean of 0 and a variance of 1. We found a significant difference between the pupil size in the street section and in the river section. The z-score had a mean of 1.0337 (median of 0.9545, variance of 0.8972) in the street section and a mean of -0.3907 (median of -0.4979, of variance 0.5559) in the river section. This is visualized in figure 5.14.

A Kolmogorov-Smirnov test indicates significantly different distributions for the samples of measured pupil size for the street and the river section ($D = 0.6313, p = 0.0$) and a two-samples t-test showed significantly lower pupil sizes in the river section than in the street section ($t(1161173) = 847.0387, p = 0.0$).

As was the case with the blink rates in section 4.1.3, we observe smaller pupil sizes in the river section, which again usually correlate with less demanding tasks and smaller cognitive load (cf. section 4.1.4). Yet, the river section should require more planning, which would also be in line with the found higher MFDs in this section.

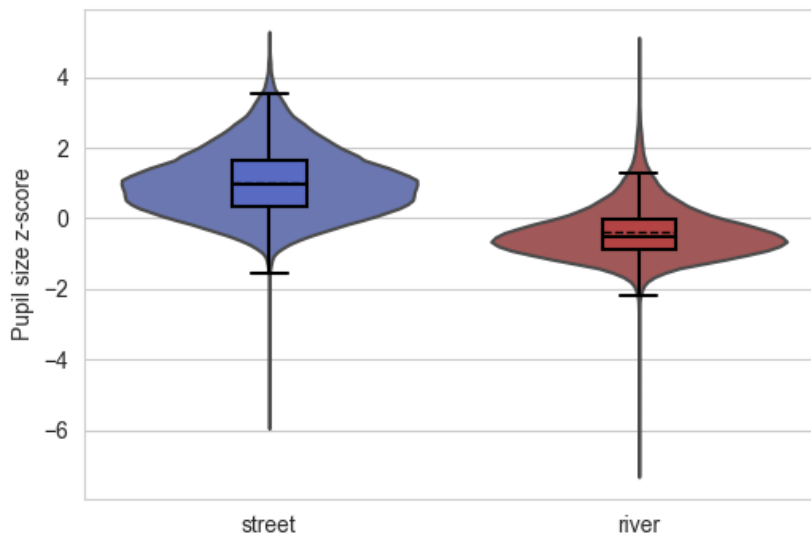


Figure 5.14.: Z-standardized pupil sizes for the street and river section ($n_{street} = 314351, n_{river} = 846824$).

5.3. Experts versus novices

To assess whether there are differences between better- and worse-performing subjects, we analyzed the same quantities as in section 5.2 but over the subjects' high scores and level scores. For example, Ma et al. (2021) found that better performing subjects in tasks have deeper planning horizons. Because of that, we expect higher MFDs, lesser blink rates and increased pupil sizes for higher scoring subjects. Section 5.3.1 analyzes the MFD, while sections 5.3.2 and 5.2.4 investigate the blink rates and pupil sizes over the subjects' scores, respectively.

At first, we tried to dichotomize the subjects into high performers and low performers. Yet, since the scores of all subjects are mostly in a similar range (cf. figure 5.3), it is difficult to find where to draw the line between these two groups. Because of that, we will rely on analyses on a continuous scale by performing linear regressions instead of comparing the means of two distinct groups.

5.3.1. Mean Fixation Distance (MFD)

Our most prominent indicator for an individual's planning horizon in our experiment is the mean fixation distance (MFD), introduced in section 4.2.

Figure 5.15 shows the average MFD of each subject over the subjects' high scores. Qualitatively, there is a visible tendency for higher scoring subjects to have higher MFDs, yet the data for high-scorers is quite sparse. A linear regression on this data shows a positive correlation ($R^2 = 0.2522$, $p = 0.0481$) with the regression function $r(x) = 1.67262 + 0.00006x$.

Another relevant measure for the correlation of planning horizon and performance could be the ability of the subjects to adapt their planning horizon accordingly between the two tasks in the street and the river section. Since we already observed higher planning horizons in the river section in section 5.2.1, we expect low differences between the MFD in the river and the street section for less-scoring subjects.

In figure 5.16, we can observe this trend. Again, as in figure 5.15, the data for high-scoring subjects is quite sparse with only two really high scoring subjects. A linear regression shows a positive correlation ($R^2 = 0.1893$, $p = 0.0787$) with the regression function $r(x) = -0.50091 + 0.00009x$, still with a less significant relation between the two quantities than in figure 5.15.

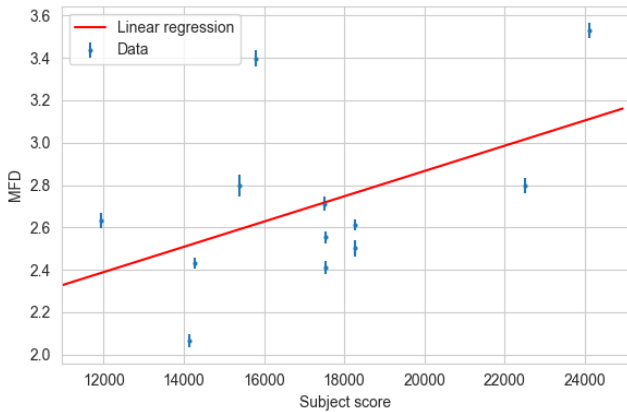


Figure 5.15.: Mean MFD of each subject over each subject's score high score with linear regression ($r(x) = 1.67262 + 0.00006x$, $R^2 = 0.2522$, $p = 0.0481$). The error bars show the standard error of the mean.

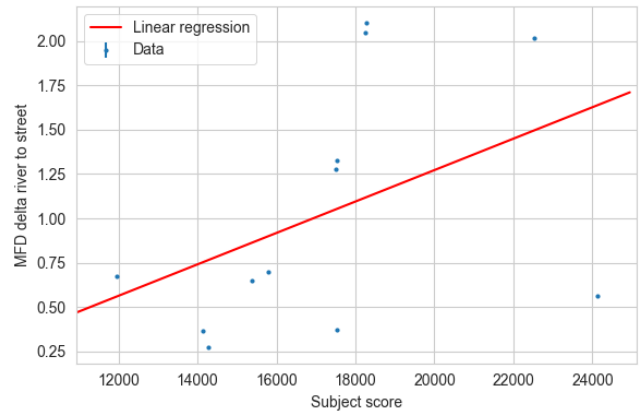


Figure 5.16.: Differences between the subjects' mean MFD in the river and street section for each subject over each subject's score high score with linear regression ($r(x) = -0.50091 + 0.00009x$, $R^2 = 0.1893$, $p = 0.0787$). The error bars show the standard error of the mean, but are barely visible.

5.3.2. Blinking behavior

We already introduced the blinking behavior in section 4.1.3 as an indicator for an individual's cognitive engagement. Cognitive load is inversely correlated with the blink rate. Since more deliberate movements in the experiment, for instance coming about by a larger planning horizon, should lead to better performances and thus higher scores, we expect lower blink frequencies for better performing subjects.

The data is visualized in figure 5.17 and we can observe that the blink rates indeed are less for the high-scoring subjects. A linear regression supports this claim, showing a negative correlation between the individual blink rates and subject scores ($R^2 = 0.2383$, $p = 0.0537$). The regression function $r(x)$ is described by $r(x) = 0.64758 - 0.00003x$.

In contrast to the unexpected increase in blink rate in the river section in section 5.2.3, our expectation were met this time. The blink rate behaves anti-proportional to the score, but proportional to the mean fixation distance, which in turn should be positively correlated with the score. This leaves us with the question, how planning processes and blink rates are correlated, which we will discuss more in-depth in section 7.1.

5.3.3. Pupil size

The pupil size is another physiological indicator of cognitive load (cf. section 4.1.4). During cognitive demanding tasks, the pupil usually dilates, resulting in a higher measurable pupil area or diameter. In accord with the relation between blink rate and performance in section 5.3.2, we expect higher pupil sizes for better performing subjects, since more successful and deeper planning strategies should need more cognitive work. The relation between pupil size and subject score is shown in figure 5.18. There is no visible correlation between the normalized pupil size and the subjects' scores ($R^2 = 0.0042$, $p = 0.8407$), with the regression function $r(x) = 0.472786 + 0.000002x$. The pupil size appears to be rather constant over all subject scores.

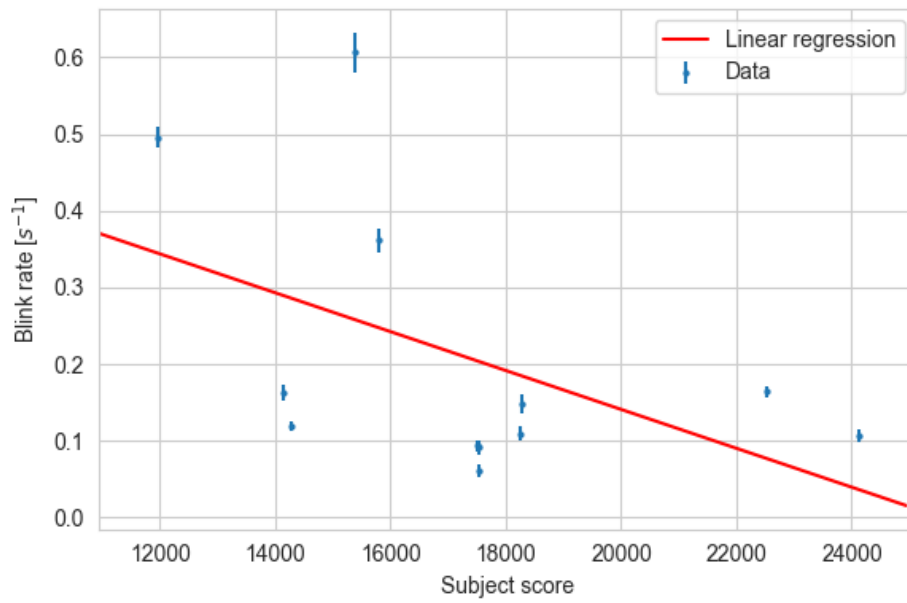


Figure 5.17.: *Blink rate as blinks per second for each subject with linear regression ($r(x) = 0.64758 - 0.00003x$, $R^2 = 0.2383$, $p = 0.0537$). The error bars show the standard error of the mean.*

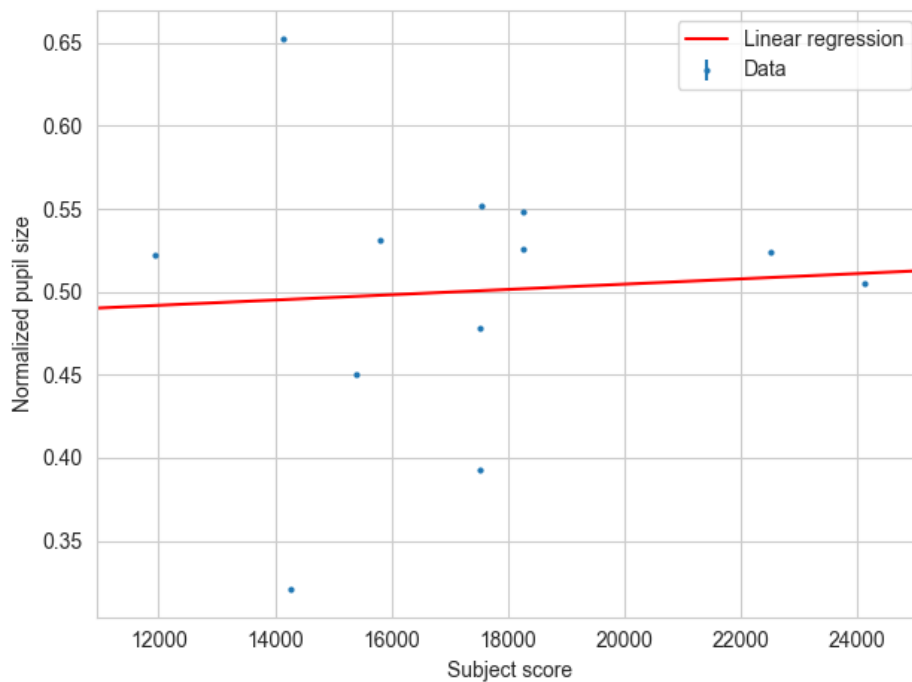


Figure 5.18.: *Normalized pupil sizes per subject over the respective subject's score with linear regression ($r(x) = 0.472786 + 0.000002x$, $R^2 = 0.0042$, $p = 0.8407$). Error bars show the standard error of the mean, but are barely visible.*

6. Gaze prediction by neural networks

Within our experiment, we gathered a lot of data, especially on the eye movements of each subject. Since we treat the eye movements and especially the fixations of each individual as externalized representations of their current planning horizon, if we can predict the amplitudes and durations of their fixations, we can also predict their planning horizon; at least to the extent that the planning horizon really is represented by the fixations. For that purpose, we implemented various different neural network architectures and trained them with the collected fixation data.

Section 6.1 introduces different ideas, which quantities can be predicted and be used as output for the neural network, as well as which data to use as input and how the different states of the environment can be represented. In section 6.2, we introduce the main architectures we used in our experiments, explain their advantages and disadvantages and why it makes sense to use this architecture particularly for our use-case. Section 6.3 shortly elaborates on the training process for the introduced networks.

6.1. Input / output data

Before we make any decisions on the design of the neural networks' architectures, we need to decide on which data to choose as input and output. Since we want to predict the planning horizon of people in various game states, we trained our neural networks with the MFDs of each fixation as output (cf. section 4.2). We also aspired to make more complex predictions, for example not only predicting the mean fixation distance in a state but also predicting the exact location that will be fixated in the current state, in either screen coordinates or field coordinates. Since this is even more challenging, we quickly turned down this idea and focused on the MFD as a single scalar output of the network only.

In this section, we elaborate more on our overall input and output data choice, as well as the structures in which we chose to represent it.

We sampled the whole state space while the subjects were playing the game, so we can provide the whole state space as input for the neural network. More complex inputs supposedly provide more information, but can contain more redundancies, while simpler inputs are more universally and easily applicable but might not be as accurate. Because of this, we also attempted to only provide the current player position in fields as input and get an accurate prediction from that. These networks quickly turned out to perform rather bad, only being able to predict the MFD with 2 fields accuracy in the validation set, while the observed values of the MFD are in the range of 0 to 10 fields.

Since we can provide the networks with the complete state, there is no real reason why we should not attempt to input the whole game state besides avoiding redundant input information.

Still, there is the matter of representation. For our experiments with the neural networks, we chose two related representations of the state. The first representation are *deep feature maps* (deep FMs) of size (15, 20,

3), where each layer represents all fields of the state of size 15×20 and represents the locations of one feature. An example of this is shown in figure 6.1. The first layer represents the position of the player, the second layer represents the position of the vehicles and the third layer represents the position of the water objects. The presence of an object on a field is represented by a 1, while the absence of an object is represented by a 0. Vehicles and water objects are present on a field if at least 50% of the field are covered by it. Since there is only one player covering only one field any time, the first layer simply contains a single 1.

A variation of the deep feature maps are *shallow feature maps* (shallow FMs), which basically are deep feature maps condensed into a single layer of size $(15, 20)$, shown in figure 6.2. Each object is represented by a distinct discrete value in this single-layer representation. The player is represented by a 1, vehicles are represented by a 2 and water objects are represented by a 3. Unoccupied fields are represented by a 0. This representation is possible since each field is occupied by one object at max, except for when the player is on top of a water object, in which case the player dominates the water object with the field being represented as occupied by the player instead of being occupied by a water object.

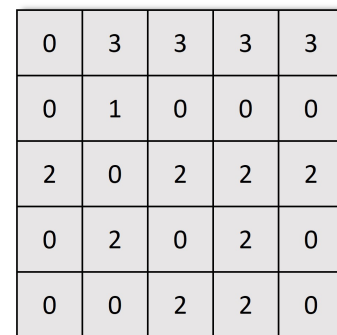
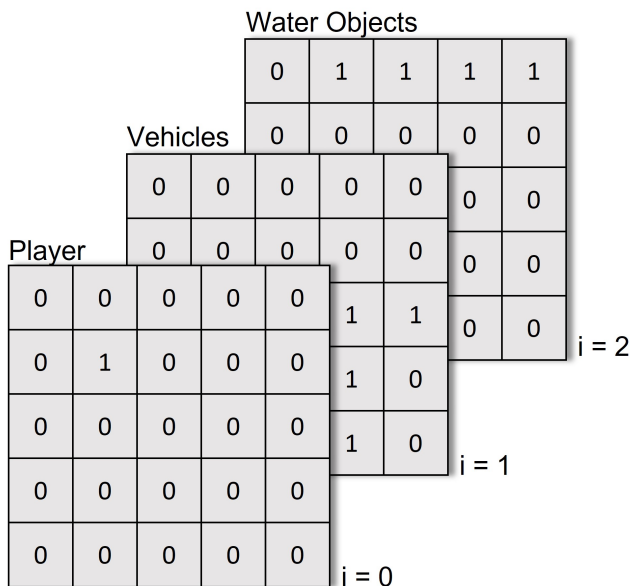


Figure 6.1.: Example of a deep feature map. Each feature is represented in its own layer as absent with a 0 or present with a 1.

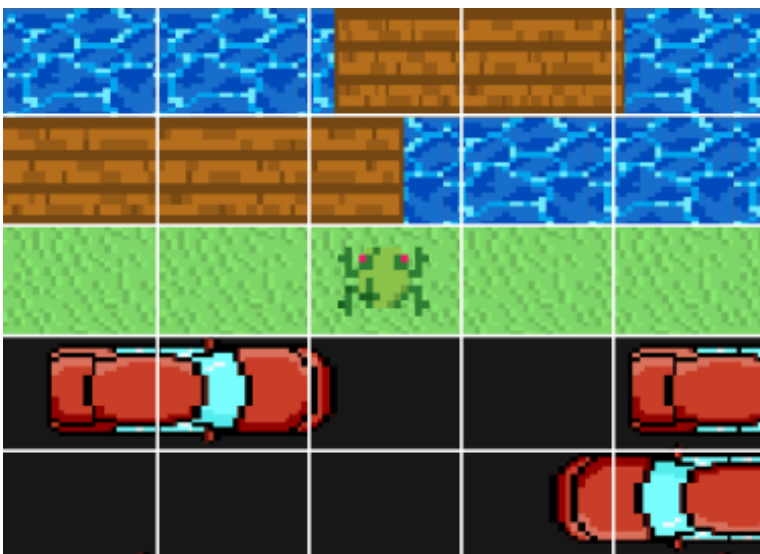
Figure 6.2.: Example of a shallow feature map. It consists only of a single layer with discrete numerical values for each feature.

Another variation of the deep and shallow FMS are what we call *avoidance maps*. These avoidance maps are basically reduced shallow FMs. An avoidance map captures the current state as a player-centered cutout of fixed size. We worked with cutouts of size 3×3 , 5×5 and 7×7 . In this cutout, each entry is either 0 or 1. A zero represents a free or 'safe' field, while a 1 represents a 'dangerous' field. We intentionally use the term 'safe' here, since, for instance, in the river section a field with a water object is occupied by an object, though only the occupied fields are safe there, because otherwise the player would touch the water. For that purpose, we did not define a field as occupied when it is covered by at least 50% of an object, but we define it as *safe* if the player could move there without colliding with a vehicle or falling into the water. An example situation is shown in figure 6.3 with its respective avoidance map representation in figure 6.4. The field in front of the player is not dangerous, since the water object covers the field such that the frog will be on top of it when

moving. The field right below the player is not represented as an occupied field, since the car is moving onto the it, but the frog will be able to move there without being driven over *immediately*. Though, it would not be clever to move there, since the car and the frog would collide before the frog is able to move on. If the player is near the edges of the screen, such that the cutout around the player also covers fields outside of the game, these fields are padded with 1s since they are not walkable, for leaving the level's boundaries will also result in a lost trial.

These avoidance maps decrease the input space a lot, while also maintaining spatial information (at least around the player, which should be of most importance). With similar definitions, avoidance maps can also generalize to other grid-like two-dimensional environments.

Still, when using avoidance maps as input, we think it is necessary to also provide the player position, since it certainly is an important feature (cf. differences between river and street, section 5.2) but can not be inferred from the avoidance map alone.



1	1	0	0	1
0	0	0	1	1
0	0	0	0	0
1	1	0	0	1
0	0	0	1	1

Figure 6.3.: Example 5×5 situation as cutout of the game 6.4.

Figure 6.4.: Example of an 5×5 avoidance map. A 0 represents a free or 'walkable' field, while a 1 represents an occupied or 'dangerous' field. The player is always at the center.

All of these state representations (deep FMs, shallow FMs, avoidance maps) can be flattened and input as a one dimensional array, yet we lose spatial information in doing so. When we input flattened states into the network, we decided on using the flattened representation of the shallow FMs, since deep FMs would make the input space really big. We did not also attempt to input the avoidance maps in a flattened version, since this modeling is rather out of the scope for this work and we could not use too much time on this. Still, this could be attempted in future work.

We also used recurrent architectures, which will be introduced in more detail in section 6.2. For now, it is only important to know that we also needed temporally structured data inputs. Each of the previously described inputs can be fed into a recurrent network as a batch of temporally preceding samples of the environment state. For example, if we want to predict the MFD at time t , we also input the last x states. Since we sampled the state with a rate of 60Hz, each two samples x_i and x_{i-1} at t_i and t_{i-1} only have a time difference of about

17ms. This usually does not provide a lot of useful information, since all objects barely move in this time frame. Because of this, we tried different numbers of time steps (for instance, feed 20, 50 or 100 previous states into the network) and also varied the ‘stride’ within the samples by always skipping a certain number of state samples before using it in the batch for the recurrent network. For example, when using a stride of 20, we only used every 20th sample of all samples from the current trial, thereby extending the time delta between the different states to about 340ms, in order to reduce redundant information by not using immediately preceding states as recurrent input.

By doing that, we also reduce the number of usable data, since there may be not enough samples early in the trials to fill the big recurrence batches, especially if we use high strides. For instance, if we use a stride of 20 and require 20 samples, we can only use data that was collected after being $20 \cdot 20 \cdot 17\text{ms} = 6.8$ seconds into the game.

6.2. Architectures

The architectures we designed are of various different forms. The simplest design only consists of classical fully-connected layers and is introduced in section 6.2.1. In section 6.2.2 we introduce our convolutional architectures, making use of spatial information in the input data. The third main variant of our architecture designs is a recurrent architecture, which requires time-series data in form of the last few states before the moment we want to have a prediction for. This architecture is shown in section 6.2.3. Section 6.2.4 contains a combination of recurrent and convolutional concepts.

We tried a lot of different architectures and variations with different activation functions, optimizers or regularizations. Still, we only include the ones here, which we did not drop immediately because of bad performance and thus became a main focus in our experiments, for example because they performed well or were modified often.

If we do not explicitly mention the used optimizer, activation function or initializer, we used the *Adam* optimizer (Kingma and Ba, 2014), rectified linear units (ReLU) as activation function (Hara et al., 2015) and *Xavier Initialization* (Glorot and Bengio, 2010).

All architectures were implemented using TensorFlow (2022).

6.2.1. Fully-connected

The first architecture we used was a simple fully connected network. It consists of one input and one output layer, with five hidden layers in between. The architecture is shown schematically in figure 6.5. Every layer was followed by a rectified linear unit as activation function and all weights were initialized using Xavier Initialization. We also experimented with different regularization methods, such as drop-out (Srivastava et al., 2014) or batch normalization (Ioffe and Szegedy, 2015).

In an adopted version, we tried only using the player position as input, in which of course the layer input was in \mathbb{R}^2 , consisting only of the x- and y-coordinate. We quickly turned down this idea in order to first find a better performing prediction of only the MFD as measure for the current planning horizon.

When using avoidance maps as inputs here, we also provide the player position by simply appending it to the flattened avoidance map input.

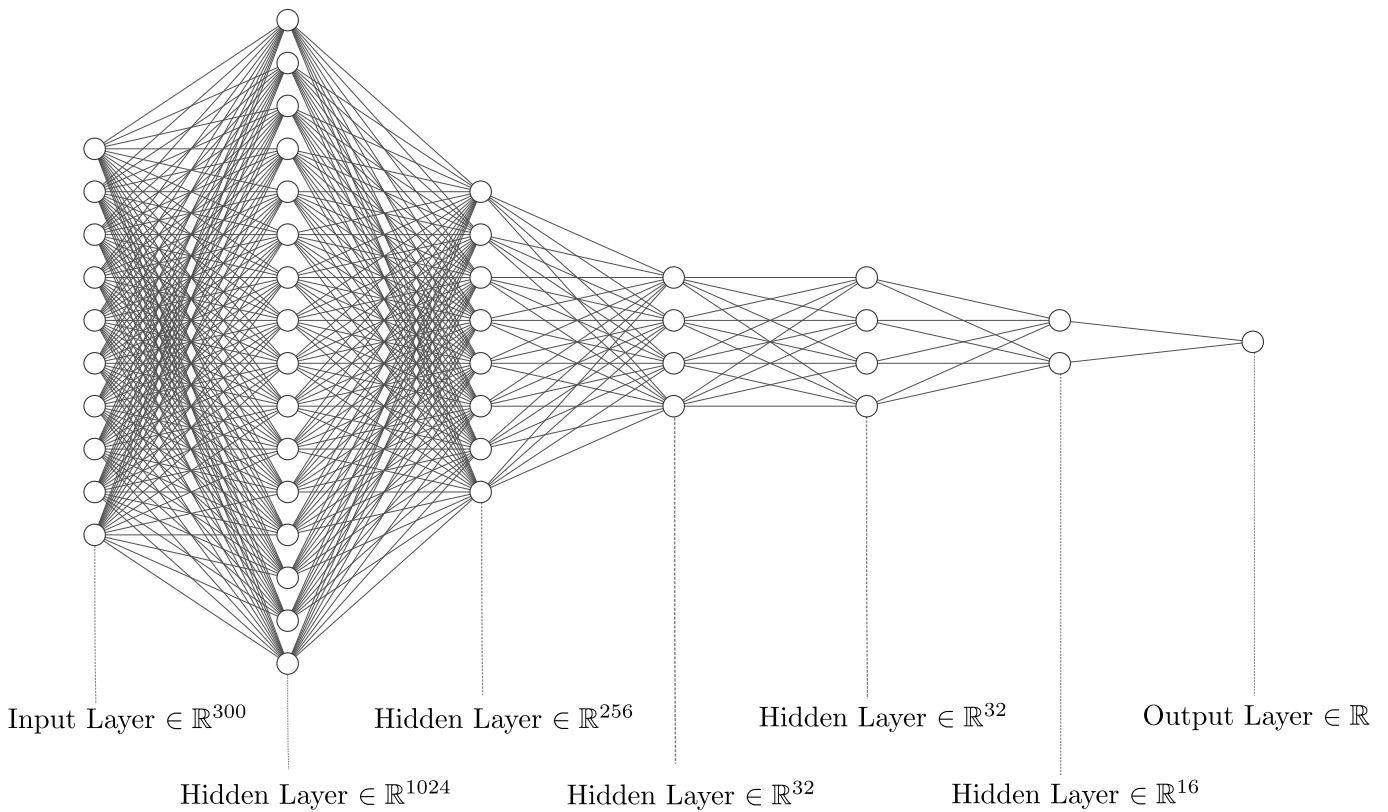


Figure 6.5.: Architecture of the fully connected network. Shown layer sizes are not to scale. Input layer for the case of a flattened shallow FM as full state input.

6.2.2. Convolutional

Since deep FMs and shallow FMs as representations of the environment state inherently contain spatial information about how the objects are currently arranged, it should benefit the prediction accuracy to make use of this. In order to do so, we also designed architectures with convolutional layers (LeCun et al., 1995), which perform a convolution with a kernel with trainable weights over an image. A deep FM can be seen as an image of size 15×20 with different channels for each object, just like a ‘real’ image would have multiple color channels, for instance RGB channels.

Our implementation consists of an input layer that adapts to the size of the chosen input, for instance a deep FM, shallow FM or an avoidance map. This is followed by two blocks of a convolution and max-pooling (Ranzato et al., 2007). The first convolution uses 64 filters of size 5×5 , followed by 2×2 max-pooling. The second convolution uses 128 filters of size 3×3 , followed by another 2×2 max-pooling, whose output is then flattened and fed to three dense layers with sizes 2560, 64, 16. All convolutional layers use a stride of 1 and padding to fit the kernel onto each layer.

A fully-connected combination of the node values of the last dense layer then forms the output.

When we used avoidance maps as input for the convolutional network, we decided it would also be beneficial to provide the player position. It cannot be inferred solely from the avoidance map, yet our experimental results have shown that it is crucial, whether the player, for instance, is in the street or the river section (cf. section 5.2).

For this purpose, we added a second input branch, which contains the player position. In this branch, the

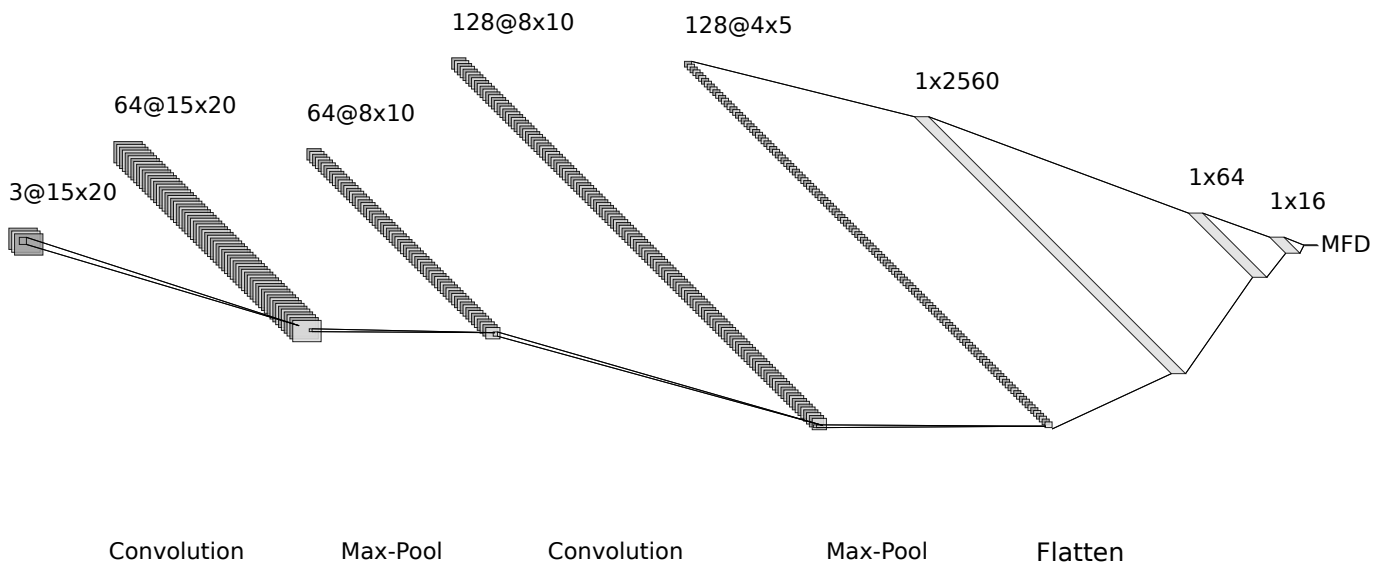


Figure 6.6.: Convolutional Architecture. Dense Layers are not to scale. This schematic is shown for a deep FM input. If a shallow FM or avoidance map is used, the input layer only consists of a single channel. Annotations at each layer show the data shape as 'channels@height×width'.

player position is first fed into a normalization layer, which transforms the player position to be centered around 0. Then, this normalized data is concatenated with the flattened output of the convolutional branch, before it is fed into the dense layers altogether. This process is visualized in figure 6.7.

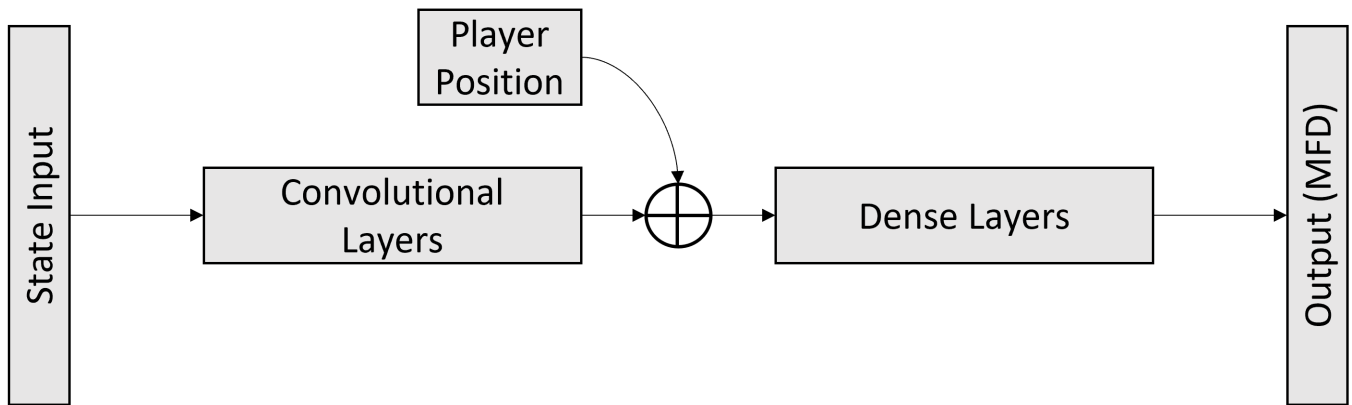


Figure 6.7.: When using the player position as supplementary input, we normalize the player position in an additional branch and add it to the flattened output of the convolutional branch, before passing it into the dense layers.

6.2.3. Recurrent / Long-Short-Term Memory

Only one sample of the environment state might not be enough to predict the current planning horizon or MFD. The planning horizon within one moment might depend on the previous actions of the player, how the objects within the level are moving, where the player fixated last and many other factors. Because of this, it might be beneficial to use a recurrent architecture, like provided by so-called *Long-Short-Term Memory*

(LSTM) cells (Hochreiter and Schmidhuber, 1997). These cells basically take a time sequence as input and process it chronologically, maintaining an internal state that is adjusted based on its previous value and the next chronological input. With that, we can extract characteristics of time sequential input data.

In our LSTM-driven model, we use one LSTM layer with 64 cells. In this case, the input is always input as a flattened shallow FM, since native LSTM layers do not support spatially structured input. The output of the LSTM layer is fed through two dense layers with 128 and 16 cells, respectively, before providing the output of the neural network. The network is shown in figure 6.8. The network may seem rather small, but the LSTM layers contain a lot of trainable parameters, which we wanted to keep in a comparable magnitude to the other network architectures, whose number of parameters is in the magnitude of about 100,000 to 120,000 trainable parameters.

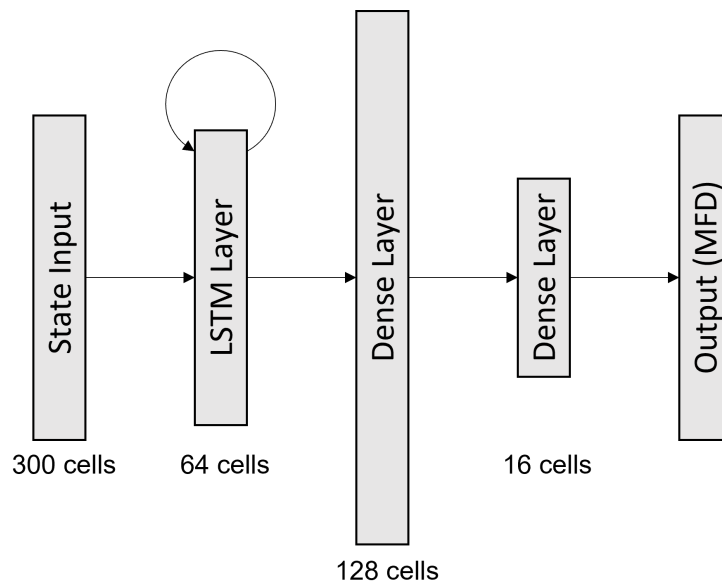


Figure 6.8.: LSTM architecture. We use one LSTM layer with 64 cells, followed by two dense layers with 128 and 16 cells, respectively. Layer sizes are not to scale.

6.2.4. Recurrent convolutional

As previously explained in section 6.2.3, native LSTM layers do not support spatially structured input, let alone convolutions. Still, Shi et al. (2015) developed a convolutional LSTM layer, which we also designed a prototype for.

The structure is very similar to the architecture of the convolutional network from section 6.2.2. It consists of an LSTM-convolutional layer with 16 filters and a filter of size 5×5 , followed by max-pooling of size 2×2 . The output of this is fed into the normal convolutional layer with 64 filters and a filter size of 3×3 , which is then flattened and fed into a dense part, consisting of three layers with 1280, 64 and 32 nodes, respectively. The architecture is shown in figure 6.9.

When using this implementation with avoidance maps as inputs, we also supply it with the player position in the same manner as shown in figure 6.7.

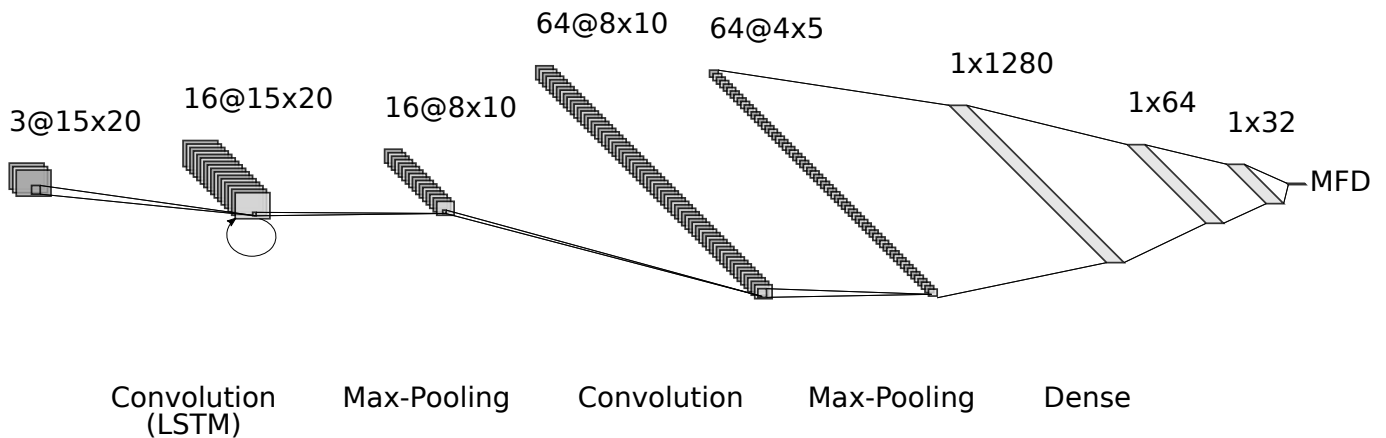


Figure 6.9.: Recurrent convolutional architecture, The first convolutional layer also employs a LSTM method as introduced by Shi et al. (2015). Dense Layers are not to scale. This schematic is shown for a deep FM input. If a shallow FM or avoidance map is used, the input layer only consists of a single channel. Annotations at each layer show the data shape as 'channels@height×width'.

6.3. Training process

We trained the shown networks with all available fixation data in a 64/16/20% training/validation/test split. We used the Adam optimizer (Kingma and Ba, 2014) for training with gradient clipping into the range $[-1, 1]$. As metric for the accuracy, we used the *root mean squared error (RMSE)*, which simplifies to the mean absolute error for single-scalar output networks. We used batches of size 32 and usually trained for 100 epochs, besides a few exceptions were training took too long and convergence could already be observed beforehand.

We trained all networks on the data we collected on all subjects. Since there may be too much aleatoric variability between all individuals, we also trained the majority of the architectures also only on the data of one subject. Since the dataset for one subject only is a lot smaller than the dataset over all subjects, we decided to train on the best performing subject, since they should provide the most data because of longer lasting trials. Because of the smaller datasets, this allowed us to train for more epochs. Therefore the subject-specific training process lasted for 200 instead of only 100 epochs. Still, this did not make a big difference, since the validation error usually already converged after 100 epochs. For some architectures with time-intensive training, like the LSTM-Convolutional architecture, we only used 50 epochs when training over all subjects because of limited resources.

Old data used in training of neural networks - Since we integrated the event detection with the *REMoDNaV* algorithm on the eye tracker samples rather late, we needed to start training the neural networks in time with fixations from a first, naïve dispersion-based fixation detection algorithm (cf. section 4.1.2). Thereby, we have more detected fixations in the data set for the neural networks than later detected by the *REMoDNaV* algorithm (Dar et al., 2021). Of course, the networks should be trained again with the new data.

The data set when training over all subjects contained 73530 fixations for non-recurrent architectures, but decreased to 20639 usable fixations in the recurrent configurations with the biggest recurrent batches. When

training only over one subject (subject G), our data set contained only 6815 fixations for non-recurrent configurations, which decreased to 1704 usable fixations for the recurrent configurations. The values of the MFD in our data set ranged from 0 to about 28.9070 fields with a mean of only 2.6991 fields (median of 2.1613 fields, variance of 4.5594 fields²); 95% of the MFDs are not bigger than 6.6396 fields.

6.4. Results

This section presents the results after training the neural networks introduced in sections 6.1 and 6.2. Section 6.4.1 gives an overview over the results of all the different architectures, while sections 6.4.2 and 6.4.3 go over the networks' performances when training over all subjects and when training for one subject only in more detail.

6.4.1. Overview

The results for all experiments with all the various architectures we designed are shown in table 6.1. Overall, the models trained with the data from all subjects are outperformed a little by the models only trained with the subject-specific data, indicating that there are individual differences that need to be considered when predicting the planning horizon.

The best performing architecture when training over all subjects was the LSTM with flat state input and its recurrent batch containing 50 previous states with a stride of 20, having a test RMSE of 1.0475 and test loss of 1.0972. This is followed by the fully connected architecture with the complete state as input, the LSTM with 100 states in the recurrent batch instead of 50 and with a stride of 20, as well as the convolutional LSTM with 7×7 avoidance maps as input. The convolutional LSTM with avoidance maps reached about the same accuracy as the convolutional LSTM with a stride of 20 and 100 previous states in the recurrent batch. The fully connected architecture provided with only the player position as input performed worst, with a test loss of 4.0662 and a test RMSE of 2.0165.

The best performing architecture in the subject-specific training for subject G was the LSTM architecture with a stride of 20 and 100 previous states in the recurrent batch, reaching a test loss of 0.6113 and a test RMSE of 0.7818. Yet, it had a much higher validation loss and RMSE in the training process, suggesting that there might be something wrong with the performance on the test set. See section 6.4.3 for more details.

The second-best performance was reached by the fully-connected architecture with avoidance maps of size 7×7 as input, followed by the fully-connected architecture with the whole (flattened) state as input and the fully-connected architecture with the 5×5 avoidance maps as input. Again, the fully-connected architecture with only the player position as input performed worst, having a test loss of 5.0382 and a test RMSE of 2.2446.

6.4.2. Results over all subjects

When training over all subjects, we reached the best performance with the LSTM architecture with flat state input and the recurrent batch containing 50 previous states with a stride of 20, having a test RMSE of 1.0475 and loss of 1.0972. Overall, the LSTM architectures performed well, even with different sizes of the recurrent batch and with the extended convolutional LSTM architecture. These results suggest that it is beneficial to also consider the recent history of states instead of only the current state. Being provided with the recent movements of all objects and the player itself, the network might be able to extract locations of special interest, which bring the highest information gain when being focused. It is also possible that the network learns which previous MFDs were most likely in the recent states and by that infer the next fixation distance more efficiently.

Still, the validation loss and RMSE at the end of the training process were larger in the convolutional LSTM setting with avoidance maps and stride 20 and size 100 in the recurrent batch, suggesting that there might

Architecture	Input (shape or parameters)	Val Loss	Val RMSE	Test Loss	Test RMSE
FC	Flat (300,)	1.3442	1.1594	1.2550	1.1203
	PP (2,)*	4.0193	2.0048	4.0662	2.0165
	AM (3, 3)*	3.7485	1.9361	3.8097	1.9518
	AM (5, 5)	2.4311	1.5592	2.5206	1.5877
	AM (7, 7)	1.8670	1.3664	1.7977	1.3408
Conv	SFM (20, 15, 1)	1.5617	1.2497	1.4477	1.2032
	DFM (20, 15, 3)	1.7702	1.3305	1.7429	1.3202
	AM (7, 7)	1.7490	1.3225	1.7824	1.3351
LSTM	Flat (s=10, size=50)	-	-	-	-
	Flat (s=20, size=50)	0.9489	0.9741	1.0972	1.0475
	Flat (s=10, size=100)	-	-	-	-
	Flat (s=20, size=100)	1.3155	1.1470	1.2956	1.1383
LSTM Conv	SFM (s=10, size=50)	-	-	-	-
	SFM (s=20, size=50)	-	-	-	-
	SFM (s=10, size=100)	-	-	-	-
	SFM (s=20, size=100)	0.8955	0.9463	1.2592	1.1221
	AM (s=20, size=100)	1.1062	1.0517	1.2586	1.1219
Architecture	Input (shape or parameters)	Val Loss (S)	Val RMSE (S)	Test Loss (S)	Test RMSE (S)
FC	Flat (300,)	0.8981	0.9477	1.0349	1.0173
	PP (2,)*	5.4449	2.3334	5.0382	2.2446
	AM (3, 3)	1.4918	1.2214	1.4748	1.2144
	AM (5, 5)	1.1278	1.0620	1.0555	1.0274
	AM (7, 7)	0.8885	0.9426	0.8140	0.9022
Conv	SFM (20, 15, 1)	1.2128	1.1013	1.3351	1.1555
	DFM (20, 15, 3)	1.3320	1.1541	1.4269	1.1945
	AM (7, 7)	1.5354	1.2391	1.7774	1.3332
LSTM	Flat (s=10, size=50)	1.1894	1.0906	1.1560	1.0752
	Flat (s=20, size=50)	1.4497	1.2040	1.1560	1.0752
	Flat (s=10, size=100)	1.4606	1.2086	1.0606	1.0299
	Flat (s=20, size=100)	1.4538	1.2057	0.6113	0.7818
LSTM Conv	SFM (s=10, size=50)	1.4223	1.1926	1.2123	1.1010
	SFM (s=20, size=50)	1.2311	1.1096	1.1794	1.0860
	SFM (s=10, size=100)	2.9015	1.7034	1.4916	1.2213
	SFM (s=20, size=100)	-	-	-	-
	AM (s=20, size=100)	1.5607	1.2493	1.7150	1.3096

Table 6.1.: Results of the different architectures with loss and RMSE on the validation set in the last iteration of training and on the test set after training. (S)-marked columns denote results from training only for subject G. The s- and size-parameters in the input column denote stride and size of recurrent batch, respectively (cf. section 6.1). Rows filled with '-' are experiments that were not run because of limited computational resources. Three best results for each category are marked in bold font. LSTM Conv with avoidance maps was run with 7×7 avoidance maps. FC=Fully-Connected, Conv=Convolutional, LSTM (Conv)=LSTM (with convolution), DFM=Deep Feature Map, SFM=Shallow Feature Map, AM=Avoidance Map, PP=Player Position, Flat=Full State (flattened); *=failed to converge or still very noisy at end of training.

be some overfitting which could be overcome with additional regularization, since we employ few to none regularization techniques in our current implementations yet.

The non-recurrent convolutional layers performed well, but did not exceed the performance of the fully-connected architecture with the flattened state input. We expected that the convolutional layers should have performed better, leveraging the spatial structure of the state representation in the shallow and deep feature maps. Yet, this does not seem to be the case. Perhaps, this is because the architecture had not enough capacity and could be improved by adding more convolutional layers or more filters in each layer.

The benefit of the convolutional layers could also be better if they are followed by a sufficiently large fully-connected block, since the current fully-connected section at the end of the convolutional networks are rather small. Perhaps, the grid of our environment is just too small with a size of 15×20 to get the usual benefits of convolutional layers. All features are rather coarse and often cover only a few cells, which might make their detection when convoluting with the convolution filters really difficult and noisy.

Worst performance was achieved by the player position only, which even failed to converge, along with the fully-connected architecture with avoidance maps of size 3×3 as input. This behavior is probably due to missing information in the input to make proper predictions. The player position alone does not suffice, showing that not only the player position in the level or the section, the player is currently in, are enough. We need to only consider the surroundings of the player and we need to do so with the necessary spatial extent, since the 3×3 avoidance maps were not enough for convergence or good performance, but with bigger avoidance maps the performance got better. Ultimately, we achieved best performances in the cases where we provided the whole state as input. Possibly, the fixation distance is also influenced by the distance to the next object, for example the fixation distance should be less when a vehicle approaches, posing a potential danger for the players and drawing their attention.

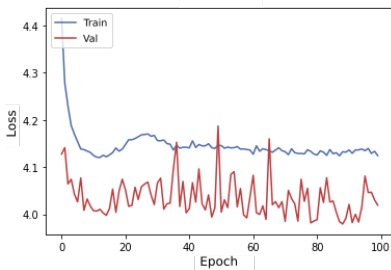


Figure 6.10.: Learning curve of the worst performing fully-connected architecture with the player position as input only. Very noisy performance on the validation set, which is even better than on the training set.

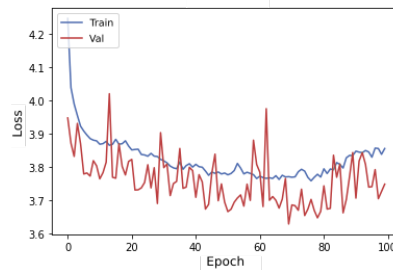


Figure 6.11.: Rather noisy learning curve of the fully-connected architecture with 3×3 avoidance maps as input, no convergence. The performance on the validation set was even better than on the training set.

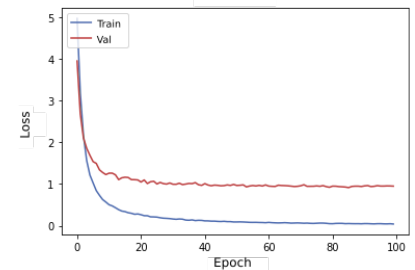


Figure 6.12.: Learning curve for LSTM architecture with a stride of 20 in the recurrent batch containing 50 previous states, which performed best. The training loss is much less than the validation loss, but they converged already after about 40 epochs.

We show some exemplary learning curves for bad performing architectures in figures 6.10 and 6.11. They do not really converge, have a rather noisy learning curve and even perform better on the validation set than on the training set. Supposedly, these predictions are mostly guided by random guessing in the correct range for the MFD and not based on any effective learning that might have happened. In contrast, the learning curve of the best performing network using LSTM layers is shown in figure 6.12. The training loss is less than the

validation loss, which is a more natural behavior, and also the learning curves already converged after about the first 40 epochs. This learning curve is qualitatively most similar to the learning curves of all other neural networks with convergent behavior as well.

6.4.3. Results for subject-specific training

For the networks that were only trained on the data of our best performing subject (subject G), the LSTM architecture with a recurrent batch containing 100 previous states with a stride of 20 performed best with a test loss of 0.6113 and a test RMSE of 0.7818. Its learning curve is shown in figure 6.14. Still, we have to be careful with this result, since the validation loss and RMSE at the end of the training process were much higher with a validation loss of 1.4538 and a validation RMSE of 1.2057.

This strong difference may suggest that the learned weights of the network simply are a lucky fit for the data in the test set. This is even more likely with the data for LSTMs with big strides and a lot of states in the recurrent batches, since there is much less data that can fill the recurrent batch, than with the non-recurrent architectures. For instance, if a trial did not last long enough, we cannot fill the whole recurrent batch and cannot use this data for our neural network (cf. section 6.1). Because of that, especially the test sets for the recurrent configurations are small. Also, the losses and RMSE on the validation and test sets usually are in about the same order of magnitude for all LSTM architectures when training subject-specific. This makes the good results with a recurrent batch containing 100 states with a stride of 20 stand out even more.

The second best performance is achieved with the fully-connected architecture, receiving avoidance maps of size 7×7 as input with a test loss of 0.8140 and a test RMSE of 0.9022. Its learning curve is shown in figure 6.15, which resembles the learning curves of most other architectures quite well, qualitatively. This time, the test loss and RMSE are also in the same magnitude as on the validation set at the end of the training process, making this result a little more trustworthy. It is still remarkable that the configuration with the avoidance map as input in the fully-connected architecture performs that well, since these configurations did not stand out at all when training over all subjects. There might be some subject-specific behavior that makes avoidance maps a better fit when training for one subject only. Yet, we do not know what this might be, besides usual between-subject differences, but it seems that the network then only needs the local surroundings of the player instead of the whole state for more accurate predictions.

Overall, the best subject-specific network results achieve a better performance than the networks which were trained on all subjects. This makes sense, since the network trained on only one subject does not average over subject-specific individualities, which might enable better predictions.

Similar to the results when training over all subjects, the convolutional architectures did not perform too bad or too well, but did not stand out in any way. Again, the capacity of the convolutional architectures might not be maxed out sufficiently or the resolution of the environment might be too coarse to make good use of convolutional layers.

We want to emphasize again, that the test sets for the recurrent architectures are already smaller than for non-recurrent architectures, not all collected data can fill the recurrent batch. This effect is even stronger when only training for one subject, since this set only contains about 1/12 of all the collected data. Our data already is not distributed very homogeneously over the state space, for example because the players tend to walk in the center of the level and not near its borders. If the data set then also becomes very small, it is more likely that it tests on the easiest to predict state samples, which were most prominent in the data set anyway.

Again, the configuration with only the player position as input performed worst with a test loss of 5.0382 and a test RMSE of 2.2446. Its learning curve is shown in figure 6.13, which was again very noisy. In contrast

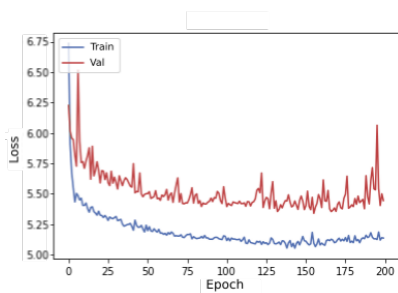


Figure 6.13.: Learning curve for the fully-connected architecture receiving only the player position as input. Very noisy and worst performance of all architectures.

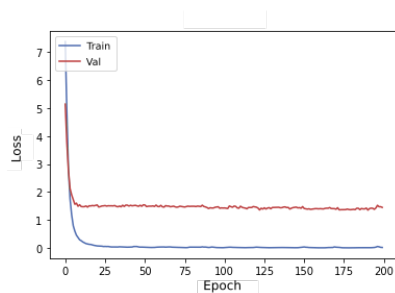


Figure 6.14.: Learning curve for the LSTM architecture with recurrent batches containing 100 previous states with a stride of 20. Best performance, though there are strong differences between the validation and the test set, which seems unusual compared to all other results.

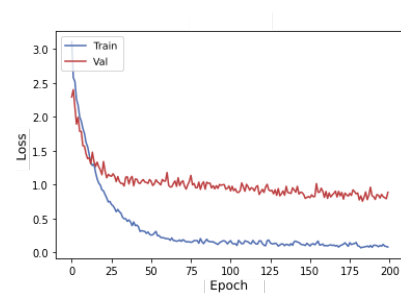


Figure 6.15.: Learning curve for the fully-connected architecture receiving 7×7 avoidance maps as input. Second best performance but with validation and test errors in about the same range.

to that, we show the learning curves of the two best performing architectures in figure 6.14 for the best performing LSTM architecture (which should still be considered with caution) and in figure 6.15 for the fully-connected architecture with 7×7 avoidance maps as inputs. Their learning curves converge after about 25 and 75 epochs, respectively.

In figure 6.16 we show the learning curve for the convolutional architecture with a recurrent batch containing 100 states and a stride of 20, because it is remarkably different from all other learning curves. While all other learning curves of networks with convergent performance continuously decrease until convergence, this learning curve does not improve for about 50 epochs, until it suddenly improves a lot and then does not change significantly in its performance. In this architecture, it seemed to tune the weights in some way which allowed for much better predictions. We could observe a similar but weaker pattern when training over all subject data with the same architecture and input.

In human terms, we would say that the network has gained some kind of insight, which we do not want to name that way since we are talking about neural networks. We do not know where this behavior originates, nor do we know what that ‘insight’ was.

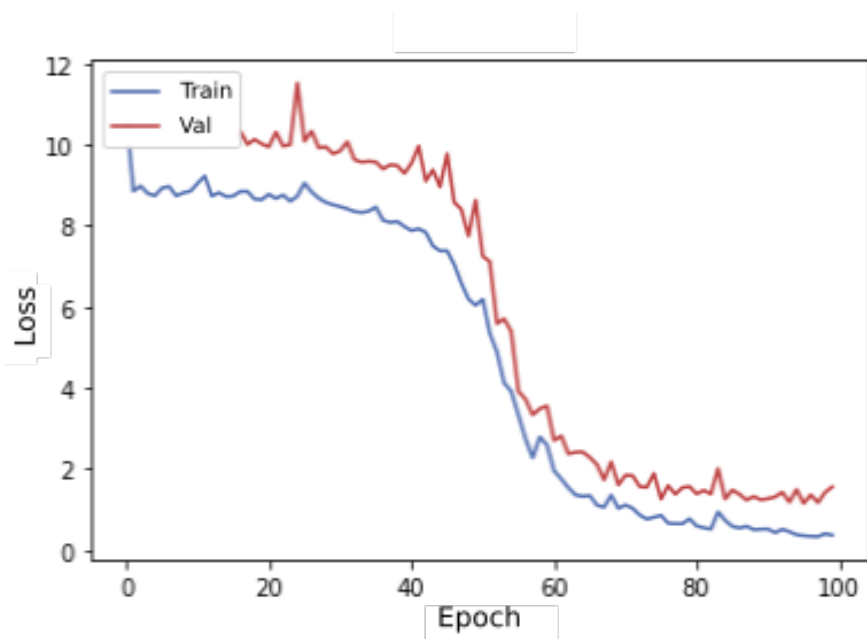


Figure 6.16.: Learning curve for the convolutional architecture with a recurrent batch containing 100 states and a stride of 20. Its weights seem to overcome a tipping point which allows for more accurate predictions after about 50 epochs.

7. Discussion

This section will discuss the results from the analysis of the experiment results (cf. section 5) in section 7.1 by summarizing the major insights and the limitations of the environment. Section 7.2 recapitulates the results of the different neural architectures when using data from all subjects or one subject only. Besides, it highlights the quality of the collected data set and the consequences for interpreting the network results from that. Section 7.3 shows perspectives on how the environment from the experiment could be modified, as well as how the neural network architectures and collected data could be extended to compensate for any shortcomings of our study.

7.1. Experiment results

The experiment results have shown two major relations: (1) The subjects were able to adapt their planning horizon dynamically when transitioning from the first task posed by the street section to the second task in the river section. The mean fixation distances and thus supposedly also the planning horizon of the subjects was significantly higher in the river section than in the street section. (2) In line with the findings of Ma et al. (2021), subjects with higher MFDs and thus also higher planning horizons performed better by successfully completing more trials in less time and thus scoring higher. In this section, we will discuss these results in more detail.

7.1.1. Humans adapt their planning horizon dynamically when switching tasks

The increased mean fixation distances in the river section suggest a higher planning horizon while the players are in the river section (cf. section 5.3.1). This makes sense, since the street section poses a classical obstacle avoidance task and the river section represents some modified version of it, since the subjects not only need to avoid obstacles (or the water), but are themselves on moving objects which do not pose an immediate threat like the vehicles in the street section do. Not only is the planning horizon increased, but also tends to be more omnidirectional instead of only extending straightly forward. This opens up the possibility for the subjects to adjust or even discard their current trajectory by also considering going backwards (cf. section 5.2.2).

The increase in MFD could arise because the subjects need longer planning horizons for the more demanding task in the river. Yet, we do not really know, if the task in the river section is really more difficult than in the street section.

The street section certainly puts more pressure on the player, since they have only limited time to plan their next move, before they have to move out of the way to not be ran over by a car. This time constraint may not allow for deeper planning and thus, higher planning horizons. In contrast, the river section does not impose any time constraint besides that the players need to avoid being carried out of the screen. This is a much

weaker constraint and leaves the players more time for planning. Still, there is a time limit on each level, forcing the players to not take too long before moving on. The level's time limit might yet be perceived less threatening, since it is not as direct a danger as the vehicles in the street section are. Also, a lot of the trials ended because of timeouts in the last lanes of the river section.

Subjects also reported that they often had to wait a long time in the river section while being on a water object already, for the next water object to appear and grant access to the next lane or the target position. The subjects might get bored and start to look around on the screen while waiting that they can take their next step, thus distorting the data set of all fixations by adding fixations that are not necessarily planning-related. Nevertheless, this seems to not be the case, since the MFDs are not only increased in the last lanes of the river section, but already at the beginning of it.

Yet overall, the subjects seem to have been able to recognize that they have the time to plan further ahead in the river section and made use of it.

According to the increased planning horizon in the river section, we should be able to observe other physiological measures of higher cognitive engagement there as well. As introduced in sections 4.1.3 and 4.1.4, decreased blink rate and increased pupil size are indicators of higher cognitive load. Yet, we could not observe any of this in the river section in comparison to the street section (cf. sections 5.2.3 and 5.2.4).

This leaves us with the question, how a longer planning horizon can come about without higher cognitive engagement. From a computational perspective, longer planning horizons create the need for more and deeper simulations for the hypothetical outcomes of taking various actions. Eventually, the perceived stress while being in the street section because of the surrounding vehicles was higher and led to a stronger activation of the sympathetic nervous system, thereby resulting in more dilated pupils and less blinking (Bradley et al., 2008). Also, studies have shown that blinks often occur at the end of cognitive processing (Siegle et al., 2008; Fukuda, 1994, 2001; Ohira, 1996), whenever working memory is released (Ichikawa and Ohira, 2004). This could correlate with the virtual simulation of one possible path in the subject's mind. If they found one path to not work out, they search for and simulate another one. Since there are many possibilities on how to cross the river section, more branches of possible paths are simulated, which could also explain the increased blink rate. Still, we have no evidence why this effect should be stronger in the river section than in the street section, except that there should be more possibilities for paths in the river section.

Besides that, this does still not explain the decreased pupil size in the river section, especially since the pupil dilates in response to blinks (Fukuda et al., 2005), which occurred more frequently in the river section. Ultimately, we could also ask if the human mind may be capable of planning further ahead and simulating more branches without more computational overhead.

7.1.2. High-performing subjects employ increased planning horizons

We could also observe that high performing subjects have higher planning horizons (cf. section 5.3.1). This is in line with the results of Ma et al. (2021). It makes sense that subjects planning further ahead are able to perform better, because they can consider and incorporate more information in the planning of their movements.

Still, we need to consider this results very cautiously, since our data for high-performing subjects is really sparse. We only had two subjects with a score higher than 20,000 points and have a big gap with no subjects in the range from 18,000 to 20,000 points. Thus, these results should be validated by collecting more data on well performing players (perhaps by especially casting experienced video game players) to get a dense and homogeneous data set on subjects and their scores.

In section 5.3.2 we examined the blink rate of the subjects in dependence with their score. We found, that higher performing subjects had smaller blink rates than lesser scoring subjects. This is in line with most of the literature, which found that higher cognitive engagement correlates with reduced blink rates, as introduced in section 4.1.3. Still, the contradiction regarding the blink rate between the river and street section persists: The decreased blink rates match the increased planning horizons for better performing subjects; yet, the planning horizon is **increased** especially in the river section, where we found **increased** blink rates.

For the pupil size, investigated in section 5.3.3, we could find no significant correlation with the subjects' scores, though we expected dilated pupils for higher performing subjects, since this usually also is an indicator of higher cognitive engagement (cf. section 4.1.4) and thus possibly also of higher planning horizons.

7.1.3. Task-specificity of the environment

Our environment contains only two very similar tasks. By the division into two sections, namely the street and river section, we also divide the environment into two different tasks, as explained in section 3.1.1. This is one strength of our setting, since we only need to focus on two different but very similar tasks. Still, this may limit our results to this specific group of tasks.

There are many approaches on how an appropriate taxonomy of human tasks could look like (e.g. Fleishman (1975); Wheaton (1968)) and there are certainly many more tasks than this kind of path finding and obstacle avoidance tasks we employ here. For example, the planning horizon may need to be assessed in a completely different way, when we are considering grasping tasks or even when we change the task from a bird's eye view to a more realistic task, solved from an ego perspective. All of this needs to be taken into account when wanting to make generalistic assumptions about the planning horizon, spanning a wider range of tasks.

7.2. Modeling with neural networks

We tested several different architectures for neural networks, finding that architectures using convolutional layers seem to have no advantages over simple fully connected architectures. Recurrent architectures seem to surpass other architectures, at least when training specifically on one subject.

The best performing networks were able to predict the planning horizon with up to one field accuracy on average when training over all subjects and up to 0.8-0.9 fields accuracy on average when training specifically on the best performing subject. Considering that 95% of the observed MFDs are in the range from 0 to about 6.6396 fields, the accuracy of our networks sets a good basis to be incrementally improved. (cf. section 6.4)

Also, the input for our networks is rather big. We need to provide the whole state in most architectures to achieve good results, since the player position does not suffice as input (worst performances). Only in the subject-specific training, we achieved good performance with avoidance maps as cutouts of the complete state. It would be simpler if we could find some key features that the networks need to perform well, to reduce the size of the necessary input data. In the best case, from that we could also conclude which features are needed to predict the human planning horizon reliably, not only for neural networks but in general.

7.2.1. Performance differences between architectures

That we did not observe an increase in performance for convolutional networks might be because of the rather small grid size of the game environment, thus having a very low resolution. Additionally, all objects in the game are maximum the height of one lane and extend over only one to four fields, but mostly two or three fields. This could make it very hard for the convolutional layers to detect and extract any of these features, resulting in no improvements in performance. It is also possible as well, that we did not provide enough capacity in the convolutional layers by, for instance, adding more layers and more filters.

The better results of the recurrent architectures make sense, since the movements of objects, for instance velocity and direction, cannot be inferred from a single state alone, but from a sequence of states. Also, how the player moved previously could already indicate how far their planning horizon has been previously and thereby also influence the size of the planning horizon the network is trying to predict in the current step. Yet, the data for training the recurrent architectures was rather small, since a lot more data is needed for filling a recurrent batch with a complete sequence of states. For example, the first few seconds of a trial cannot be used as training data for recurrent architectures with large recurrent batches, since the preceding course of the game is still too small to fill the recurrent batches completely. More data could even increase their performance more and provide larger and more reliable test data sets. Furthermore, the outstanding performance of the best LSTM in the subject-specific training has to be considered with caution, since it performs way better on the test set than on the validation set during training, which seems highly unusual, since it should rather perform worse on the unseen data from the test set.

7.2.2. Neural networks perform better on single-subject data

The performance of the networks is overall better when training only on the data of one subject. Since all people will probably have some individual characteristics in how far and in what way their planning horizons extend, this is only logical. When we are using the data from all subjects for training, these individual aspects get lost and we are smoothing over individual traits. In practice, it would be beneficial to have one network which can make predictions for anybody and not only for one specific individual. Also, we have less data to train on if we train on one subject only and are also required to already have been able to gather data on this specific individual.

7.2.3. Insights from neural network results

From the worse results when only providing the player position as input, we can learn which features are needed to make good predictions. Since we always needed to provide the full state as input or at least a state covering a sufficiently large area around the player, we can assume that some of the data contained in these states is needed for accurate predictions of the planning horizon. Even the avoidance maps, which only cover the area around the player, but still achieve good results in the subject-specific training (cf. section 6.4.3), must contain important factors influencing the planning horizon, at least for our best performing subject.

It is likely that the planning horizon is strongly influenced by the objects in the immediate surroundings of the players, especially in the street section, where the vehicles are posing a direct threat to the player. Since this information is contained in the full environment states and sufficiently big avoidance maps, the distance to the next objects is perhaps a key factor on how far the planning horizon extends. When the players employ larger planning horizons, they need to redirect their attention farther away from their current position and

thus lose track of the objects around them. This is not the case if they stick to smaller planning horizons, making smaller planning horizons more safe in the street section.

7.2.4. Homogeneity and structure of data

We collected a lot of data in our experiment. Still, this data is not distributed very homogeneously, since we mostly collected states where the players moved near the horizontal center of the environment, since this is the most safe strategy for them. Assuming that the player can be at any of the $14 \cdot 20 = 280$ positions of the environment and each of the fields in the six lanes of each, river and street, section can be covered by an object or not, we have a rough estimate of about $2^{12 \cdot 20} = 2^{240}$ possible arrangements of objects. This is only an upper bound, since we neglect that the player cannot stand on water or on a car and the patterns in which objects are spawned. Yet, the idea is clear, that by this estimate there are $280 \cdot 2^{240}$ possible states. Considering that our data only comprises 73,530 fixations, this is not even close to covering the whole state space. Also, we have much more fixations in the river section (45,156) than in the street section (14,982). Thus, the data is not only inhomogeneous regarding the horizontal position of the player, but also regarding the section the player was in. For accurate predictions for any kind of input, we should strive for dense, homogeneous and more data.

If we assume that not the full state is needed but only the immediate vicinity of the player, the needed data for covering various states reduces a lot. If we consider avoidance maps of size 7×7 as sufficient input, our state space would consist of roughly $2^{7 \cdot 7} = 2^{49}$ possible states, which is much less, but still much bigger than our data set.

This course of dimensionality even gets worse, as soon as we use recurrent data. Since we now also have an additional, temporal dimension in our data, we have much less data points to train on. We can only fill recurrent batches with data from trials which lasted long enough. Moreover, if we use recurrent batches containing a 100 previous states as input, each of these 100 states can have one of all possible state configurations. In practice, the dynamics between the states are constrained by time, because at least in our environment all changes in object positions should be rather smooth. Still, recurrent architectures need even more data for appropriate training.

Another aspect is the way we structure our data. We introduced avoidance maps as reduced state representation, containing only the immediate vicinity of the player. Since they do not comprise the raw state, but rather a processed version of it, denoting which fields the player can step on without losing the trial, we might go too far with this already. By taking this step, we may take away information which is important for the network or aspects that have influenced the players' planning horizons. For instance, the distinction between river and street objects is lost within the avoidance map representation. The architectures should also be tested with the raw state input, for example by a SFM or DFM, only reduced in size to the vicinity of the player, without any other processing.

7.3. Future work

During and after the analysis of our results, we realized which ways would be optimal to extend our research and identified possible shortcomings of our implementation. This section will elaborate on the overall experimental setup and how it can be extended, as well as how our neural network models can be improved and which possibilities they open up for further possible insights on the human planning horizon.

7.3.1. Experiment variants

In our analysis, we already see various tendencies in the data, for example that the planning horizon is higher in the river section or that better performing subjects employ bigger planning horizons. Especially in the latter analysis, we do not have enough data to completely rely on the results of the linear regression.

Our current results suggest a significance relation between performance and planning horizon, yet our data is not as well distributed over the range of possible scores as we would like it to. From the originally 16 subjects we cast, we needed to remove the data from four of them because of technical difficulties, like the eye tracker not working reliably or flaws in the implementations when collecting data on the first subjects. In the remaining 12 subjects, we have two very good performing subjects with scores above 22000 points, but the other subjects all have a score in the range [12000, 19000], leaving a rather big gap between them. If the increased planning horizons observed in our well-performing subjects are noise or outliers, we are not able to identify them as such in our analysis. This gap in the data could be filled by collecting data on some more subjects, maybe especially inviting experienced gamers, which should have a higher chance of performing well and reaching higher scores.

Our findings suggest that players employ bigger planning horizons in the river section (cf. section 5.2.1), but still this could be explained by several other factors. For example, the planning horizon could be less in the street section since the vehicles do not leave as much time to plan ahead as the water objects in the river section do, where the players can sit on until they reach the left or right boundary of the level (cf. section 7.1.1).

We could test this hypothesis by controlling the environment more strongly, for example by adding a feature of the original Frogger to the river section, where water objects randomly submerge, thereby forcing the player to leave. This would reduce the time the players can stay on the water objects and should be controlled such that the time between having to move out of the way of a car and the time before moving onto another water object is the same on average. We could then test, whether the planning horizon is still bigger in the river section, even if the time enforcing the next action is the same in both sections.

Even though we found differences in the planning horizons by means of the mean fixation distances of the subjects, we could not find any physiological evidence for higher cognitive loads when employing bigger planning horizons (cf. sections 5.2.3 and 5.2.4). For testing this, we should first collect more data and see whether this holds to be true. If this is the case, more experiments should be designed, especially testing similar versions of the tasks and environment used here, to investigate under which conditions people are able to plan ahead without any cognitive overhead. It is rather unlikely that people have no computational overhead when using larger planning horizons, yet this should be tested to explain or invalidate our results in any way.

Since we were able to find differences in the planning horizons when solving different tasks, this could be tested for more other tasks than only the two tasks from our experiment as well. For that, a well-defined task taxonomy would be helpful, such that new experiments can be designed in a way that they can identify

quantitative differences in planning horizon over a wide range of different tasks. On the other hand, the planning horizon of individuals while solving different tasks could also set the basis for a new taxonomy based on the extent of people’s planning horizon.

Subject feedback on game design

We received a lot of subject feedback on the general dynamics and appearance of the game, which should be incorporated when collecting data in the future.

First of all, a lot of subjects criticized the hit boxes of the vehicles in the street section as too harsh. We chose the hit boxes such that they do not extend over the textures’ limits. Most likely, this choice is not necessarily bad, but the extent of the hit boxes is not easy to recognize from the textures. While the cars are solid and good to see obstacles, the legs of the frog representing the player are only a few pixels big and may not be appropriate, since a car slightly touching the frog’s legs already leads to a lost trial. This could be overcome by choosing another sprite for the player and if this does not help, the hit boxes should be adjusted to a more ‘relaxed’ version.

Moreover, some subjects described the last lanes of the river section as a little too difficult. Not in the sense, that it was too hard to maneuver the frog over the water objects, but that especially in more difficult levels the water object spawned too rarely, resulting in the players having to wait a long time before there even opened up a possibility to access the target position. This also shows in figure 5.4, showing the last river lanes having a lot more time outs than the lanes before. Of course, it is to expect that more timeouts happen in the latter lanes, since more time passes until the players reach these lanes. Yet, the figure shows a rather distinct peak in the number of timeouts in these last lanes. This can be easily adjusted by tuning the generative model for the levels and increasing the spawn probability θ or decreasing the distance $d_{\text{water object}}$ between the water objects (cf. table 3.1), especially for latter lanes.

Some subjects also remarked that they did not feel like they could plan ahead a lot and also did not give feedback that they employed a certain strategy, when asked for it in the assessment after the experiment. Since we clearly found differences in the planning horizon in different regions of the game, they must have had a change in their strategy, but probably did not experience this consciously. Still, the level sizes could be increased in future experiments, thereby leaving more space to plan.

7.3.2. Extending experiments with neural networks

Since the prediction of planning horizons was not the main focus of this work and it could easily cover multiples studies on its own, there are plenty of possibilities to extend and improve this.

First of all, the different types of architectures could be tested with varied capacity. For example, the convolutional architectures did not bring the expected increase in performance, which could be tested again with more convolutional layers, containing more and different kernels. The optimal input for recurrent architectures in terms of which and how many state samples are provided each time, could also further improve performance. Overall, our neural networks lack proper regularization methods, which should be added.

The architectures in their current form mostly rely on the full state to achieve good performance. The player position as only input did not suffice and only in the subject-specific training the avoidance maps as a reduced form of the full states led to good predictions. If we could identify key features that are essential for predicting the planning horizon reliably, this could simplify our models and lead to more accurate predictions. The fact

that the avoidance maps as input led to better performances in the subject-specific training than the full state, suggests that local features can provide more information relevant for the prediction, at least if we do not average over multiple subjects. Potentially, we could make use of this by providing further information on the subjects as input to the network, for example their achieved maximum score or average level score, to give the network the possibility to identify the key characteristics of each subject's planning behavior. Thereby, we could possibly leverage all of the data in our data set with higher accuracy instead of only using the data from one subject.

Besides that, the input data to the recurrent architectures could also consist of the previous fixations or previous fixation distances as well, which could maybe turn out as more important than the actual current game state. If the player previously scanned a lot of their immediate vicinity, they should be more likely to fixate on areas farther away from them.

In general, the lack of data also applies here. Not in the sense that we do not have enough subjects, but that we need more fixations in different states, for example not only when the subjects are moving near the horizontal center of the level, which they usually did the most. This could be encouraged by awarding bonus points for moving near the edges, though this already enforces certain behavior. This could also be reduced by identifying the key features needed for good predictions, which could be accomplished by applying principal component analysis or similar methods to the collected data.

As soon as we are able to make good predictions for the weighted fixation distance, we could try to extend our network by maybe predicting the fixation distance and duration distinctly or trying to predict the exact position or field that will be fixated next. These extensions also open up the possibility to integrate this predictor in a bigger system, for instance in an optimal control (Lewis et al., 2012) or reinforcement learning setting. The eye movements could be modeled by the network and embedded into a reinforcement learning agent, whose behavior is described with a POMDP (Spaan, 2012; Kaelbling et al., 1998), whose observations are guided by the predictions of the eye movement predictor. Thus, in the optimal case the reinforcement learning agent would be provided with the same kind and amount of information as it is the case for real humans to decide its next action.

Moreover, our current models can handle only input from our game. Still, the representation of game states that rely on a discrete, grid-like structure, could be generalized to other games as well. For example, the different layers of a deep feature map could represent the player in one layer, fields to avoid in another layer, target fields in another layer and so on. By this, we could also train and use our models on data from other games and tasks, as soon as we found a well-working model for our scenario.

Learning about the human planning horizon from models

If we collected more data, adjusted our architectures a little more and finally achieved good performance on most of the state space, we could also learn more about the human planning horizon from the results and behavior of the model. Our current results from the subject-specific training with avoidance maps as input already suggest that local features might be more important for predictions than global features. Since neural networks only are black-box models and do not provide further insights in how this comes about, we can only hypothesize why this is the case. Yet, it supports our hypothesis that fixation distances are less in the street section, since the vehicles pose local threats to the players, leaving them less time to plan further ahead.

By simulating a lot of different of scenarios, in which the player is more or less far away from dangerous objects, we could also try to define a cost function which trades off the potential information gain from performing a fixation to an area farther away versus the (perceived) time that is left before the player must take action to move out of the way of the object.

Generally, our current analysis only allows us to get an estimate of the player's current planning horizon every time they are performing a fixation. If we gather enough data on the different states in which various fixations are performed, our model may be able to generalize to make accurate predictions of the planning horizon in any state, not only in the states in which an actual fixation is performed by the player. This would allow for an more in-depth analysis of the variation of the planning horizon at any point in time. For example, how fast or smooth it changes. With the (estimated) planning horizon in each state of the game at hand, identification of features which trigger changes in the planning horizon also becomes a lot easier. Yet, exactly these features might be necessary to create a model which provides realistic results at all.

8. Conclusion

We designed a *Frogger*-like environment, in which players must find a path from the bottom of a screen to the top of the screen while avoiding obstacles. The environment contains two different but similar obstacle avoidance tasks, in each of which the player must plan sufficiently ahead to perform well. Players are rewarded with bonus points if they manage to solve a level of the game in less time.

We collected data on 12 subjects, also measuring their eye movements while planning. By treating the collected eye movement data as an externalization of their planning horizon, we can estimate the current planning horizons of the subjects. Our analyses have shown that people are able to dynamically adjust the spatial extent and direction of their planning horizon when switching between tasks. Moreover, our results suggest that subjects with higher planning horizons perform better. Still, we could not find any physiological indicators for higher cognitive engagement when employing bigger planning horizons, but even observed the contrary: subjects' blink rates and pupil sizes suggested lesser cognitive loads while solving the task that was solved with a smaller planning horizon.

To further leverage the collected data, we designed various neural networks and tested several different kinds of architectures and combinations of input and output data. We were able to predict the subjects' fixation distances with about 1 field accuracy when training with the data of all subjects. When training for only one subject, we achieved an accuracy of 0.6 to 0.8 fields.

Future work should focus on how the missing relation between planning horizon and cognitive load in our case came about. The environment can be extended to other tasks as well, in order to find qualitative differences in planning horizon over a wide range of tasks. Improvements of the neural networks for predicting the planning horizon more accurately and in other tasks as well would allow for a more in-depth analysis on how the planning horizon of subjects changes in different situations and progresses over time.

References

- Attard-Johnson, J., Ó Ciardha, C., and Bindemann, M. (2019). Comparing methods for the analysis of pupillary response. *Behav Res*, 51(1):83–95.
- Beatty, J. (1982). Task-evoked pupillary responses, processing load, and the structure of processing resources. *Psychological Bulletin*, 91(2):276–292.
- Bradley, M. M., Miccoli, L., Escrig, M. A., and Lang, P. J. (2008). The pupil as a measure of emotional arousal and autonomic activation. *Psychophysiology*, 45(4):602–607.
- Bristow, D., Haynes, J.-D., Sylvester, R., Frith, C. D., and Rees, G. (2005). Blinking Suppresses the Neural Response to Unchanging Retinal Stimulation. *Current Biology*, 15(14):1296–1300.
- Callaway, F., Jain, Y. R., van Opheusden, B., Das, P., Iwama, G., Gul, S., Krueger, P. M., Becker, F., Griffiths, T. L., and Lieder, F. (2022a). Leveraging artificial intelligence to improve people’s planning strategies. *Proceedings of the National Academy of Sciences*, 119(12):e2117432119. Publisher: National Acad Sciences.
- Callaway, F., van Opheusden, B., Gul, S., Das, P., Krueger, P., Lieder, F., and Griffiths, T. (2021). Human planning as optimal information seeking. *Manuscript in preparation*.
- Callaway, F., van Opheusden, B., Gul, S., Das, P., Krueger, P. M., Griffiths, T. L., and Lieder, F. (2022b). Rational use of cognitive resources in human planning. *Nat Hum Behav*, 6(8):1112–1125.
- Carpenter, R. H. (1988). *Movements of the Eyes, 2nd Rev.* Pion Limited.
- Carton, D., Nitsch, V., Meinzer, D., and Wollherr, D. (2016). Towards Assessing the Human Trajectory Planning Horizon. *PLoS ONE*, 11(12):e0167021.
- Dalveren, G. G. M. and Cagiltay, N. E. (2019). Evaluation of ten open-source eye-movement classification algorithms in simulated surgical scenarios. *IEEE Access*, 7:161794–161804. Publisher: IEEE.
- Dar, A. H., Wagner, A. S., and Hanke, M. (2021). REMoDNaV: robust eye-movement classification for dynamic stimulation. *Behav Res*, 53(1):399–414.
- Dolan, R. and Dayan, P. (2013). Goals and Habits in the Brain. *Neuron*, 80(2):312–325.
- Doll, B. B., Duncan, K. D., Simon, D. A., Shohamy, D., and Daw, N. D. (2015). Model-based choices involve prospective neural activity. *Nat Neurosci*, 18(5):767–772.
- Drew, G. C. (1951). Variations in Reflex Blink-Rate during Visual-Motor Tasks. *Quarterly Journal of Experimental Psychology*, 3(2):73–88.
- Dye, M. W., Green, C. S., and Bavelier, D. (2009). Increasing Speed of Processing With Action Video Games. *Curr Dir Psychol Sci*, 18(6):321–326.
- Fatt, I. and Weissman, B. A. (2013). *Physiology of the eye: an introduction to the vegetative functions*. Butterworth-Heinemann.

-
- Findlay, J. M., Findlay, J. M., Gilchrist, I. D., and others (2003). *Active vision: The psychology of looking and seeing*. Number 37. Oxford University Press.
- Fleishman, E. A. (1975). Toward a taxonomy of human performance. *American Psychologist*, 30(12):1127–1149.
- Fukuda, K. (1994). Analysis of eyeblink activity during discriminative tasks. *Perceptual and motor skills*, 79(3_suppl):1599–1608. Publisher: Sage Publications Sage CA: Los Angeles, CA.
- Fukuda, K. (2001). Eye blinks: new indices for the detection of deception. *International Journal of Psychophysiology*, 40(3):239–245. Publisher: Elsevier.
- Fukuda, K., Stern, J. A., Brown, T. B., and Russo, M. B. (2005). Cognition, blinks, eye-movements, and pupillary movements during performance of a running memory task. *Aviation, space, and environmental medicine*, 76(7):C75–C85. Publisher: Aerospace Medical Association.
- Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterton, M., editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.
- Green, C. S. and Bavelier, D. (2003). Action video game modifies visual selective attention. *Nature*, 423(6939):534–537.
- Griffiths, T. L., Callaway, F., Chang, M. B., Grant, E., Krueger, P. M., and Lieder, F. (2019). Doing more with less: meta-reasoning and meta-learning in humans and machines. *Current Opinion in Behavioral Sciences*, 29:24–30.
- Halan, S., Rossen, B., Cendan, J., and Lok, B. (2010). High score!-motivation strategies for user participation in virtual human development. In *International conference on intelligent virtual agents*, pages 482–488. Springer.
- Hara, K., Saito, D., and Shouno, H. (2015). Analysis of function of rectified linear unit used in deep learning. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.
- Hayhoe, M. and Ballard, D. (2005). Eye movements in natural behavior. *Trends in Cognitive Sciences*, 9(4):188–194.
- Hess, E. H. and Polt, J. M. (1964). Pupil Size in Relation to Mental Activity during Simple Problem-Solving. *Science*, 143(3611):1190–1192.
- Hessels, R. S., Niehorster, D. C., Nyström, M., Andersson, R., and Hooge, I. T. (2018). Is the eye-movement field confused about fixations and saccades? A survey among 124 researchers. *Royal Society open science*, 5(8):180502. Publisher: The Royal Society.
- Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.
- Holland, M. K. and Tarlow, G. (1972). Blinking and Mental Load. *Psychol Rep*, 31(1):119–127.
- Holland, M. K. and Tarlow, G. (1975). Blinking and Thinking. *Percept Mot Skills*, 41(2):403–406.
- Ichikawa, N. and Ohira, H. (2004). Eyeblink activity as an index of cognitive processing: Temporal distribution of eyeblinks as an indicator of expectancy in semantic priming. *Perceptual and Motor Skills*, 98(1):131–140. Publisher: SAGE Publications Sage CA: Los Angeles, CA.

-
- Ioffe, S. and Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France. PMLR.
- Jain, Y. R., Callaway, F., Griffiths, T. L., Dayan, P., He, R., Krueger, P. M., and Lieder, F. (2022). A computational process-tracing method for measuring people’s planning strategies and how they change over time. *Behav Res.*
- Jones, J. L., Esber, G. R., McDannald, M. A., Gruber, A. J., Hernandez, A., Mirenzi, A., and Schoenbaum, G. (2012). Orbitofrontal Cortex Supports Behavior and Learning Using Inferred But Not Cached Values. *Science*, 338(6109):953–956.
- Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Land, M. and Tatler, B. (2009). *Looking and acting: vision and eye movements in natural behaviour*. Oxford University Press.
- Landers, R. N., Bauer, K. N., and Callan, R. C. (2017). Gamification of task performance with leaderboards: A goal setting experiment. *Computers in Human Behavior*, 71:508–515.
- LeCun, Y., Bengio, Y., and others (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995. Publisher: Cambridge, MA USA.
- Lee, M. D., Gluck, K. A., and Walsh, M. M. (2019). Understanding the complexity of simple decisions: Modeling multiple behaviors and switching strategies. *Decision*, 6(4):335–368.
- Lee, P. F. and Lee, K. S. (2017). A PRELIMINARY STUDY ON PUPIL SIZE CHANGES DETECTION ON MENTAL STRESS WITH MENTAL STRESSOR. *Biomed. Eng. Appl. Basis Commun.*, 29(02):1750011.
- Lewis, F. L., Vrabie, D., and Syrmos, V. L. (2012). *Optimal control*. John Wiley & Sons.
- Ma, I., Ma, W. J., and Gureckis, T. M. (2021). Information sampling for contingency planning. In *Proceedings of the 43rd Annual Conference of the Cognitive Science Society*.
- MacIver, M. A., Schmitz, L., Mugan, U., Murphey, T. D., and Mobley, C. D. (2017). Massive increase in visual range preceded the origin of terrestrial vertebrates. *Proc. Natl. Acad. Sci. U.S.A.*, 114(12).
- Medathati, N. V. K., Desai, R., and Hillis, J. (2020). Towards inferring cognitive state changes from pupil size variations in real world conditions. In *ACM Symposium on Eye Tracking Research and Applications*, pages 1–10, Stuttgart Germany. ACM.
- Meder, B., Nelson, J. D., Jones, M., and Ruggeri, A. (2019). Stepwise versus globally optimal search in children and adults. *Cognition*, 191:103965.
- Mugan, U. and MacIver, M. A. (2020). Spatial planning with long visual range benefits escape from visual predators in complex naturalistic environments. *Nat Commun*, 11(1):3057.
- Nelson, J. D., Meder, B., and Jones, M. (2018). Towards a theory of heuristic and optimal planning for sequential information search.

-
- Nishizono, R., Saijo, N., and Kashino, M. (2021). Synchronization of Spontaneous Eyeblink during Formula Car Driving. In *ACM Symposium on Eye Tracking Research and Applications*, pages 1–6, Virtual Event Germany. ACM.
- Nyström, M. and Holmqvist, K. (2010). An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. *Behavior Research Methods*, 42(1):188–204.
- Ohira, H. (1996). Eyeblink activity in a word-naming task as a function of semantic priming and cognitive load. *Perceptual and motor skills*, 82(3):835–842. Publisher: SAGE Publications Sage CA: Los Angeles, CA.
- Pedersen, M. K., Rasmussen, N. R., Sherson, J. F., and Basaiawmoit, R. V. (2017). Leaderboard Effects on Player Performance in a Citizen Science Game.
- Peng, X.-W., He, Q.-C., Ji, T., Wang, Z.-L., and Yang, L. (2006). Mental workload for mental arithmetic on visual display terminal. *Zhonghua lao Dong wei Sheng zhi ye Bing za zhi = Zhonghua Laodong Weisheng Zhiyebing Zazhi = Chinese Journal of Industrial Hygiene and Occupational Diseases*, 24(12):726–729.
- Pohlen, T., Piot, B., Hester, T., Azar, M. G., Horgan, D., Budden, D., Barth-Maron, G., van Hasselt, H., Quan, J., Večerík, M., Hessel, M., Munos, R., and Pietquin, O. (2018). Observe and Look Further: Achieving Consistent Performance on Atari.
- Poulton, E. C. and Gregory, R. L. (1952). Blinking during Visual Tracking. *Quarterly Journal of Experimental Psychology*, 4(2):57–65.
- Ranzato, M. a., Boureau, Y.-l., and Cun, Y. (2007). Sparse Feature Learning for Deep Belief Networks. In Platt, J., Koller, D., Singer, Y., and Roweis, S., editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc.
- Rayner, K. (2009). The 35th Sir Frederick Bartlett Lecture: Eye movements and attention in reading, scene perception, and visual search. *Quarterly journal of experimental psychology*, 62(8):1457–1506. Publisher: SAGE Publications Sage UK: London, England.
- Redish, A. D. (2016). Vicarious trial and error. *Nat Rev Neurosci*, 17(3):147–159.
- Savitzky, A. and Golay, M. J. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639. Publisher: ACS Publications.
- Schiffman, H. R. (1990). *Sensation and perception: An integrated approach*. John Wiley & Sons.
- Schmidt, B., Duin, A. A., and Redish, A. D. (2019). Disrupting the medial prefrontal cortex alters hippocampal sequences during deliberative decision making. *Journal of Neurophysiology*, 121(6):1981–2000.
- Schwitzgebel, E. (2008). The Unreliability of Naive Introspection. *The Philosophical Review*, 117(2):245–273.
- Shapiro, K., Raymond, J., and Arnell, K. (1997). The attentional blink. *Trends in Cognitive Sciences*, 1(8):291–296.
- Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-k., and WOO, W.-c. (2015). Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.
- Siegle, G. J., Ichikawa, N., and Steinhauer, S. (2008). Blink before and after you think: Blinks occur prior to and following cognitive load indexed by pupillary responses. *Psychophysiology*, 45(5):679–687.

-
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489.
- Şimşek, Ö., Algorta, S., and Kothiyal, A. (2016). Why most decisions are easy in tetris—and perhaps in other sequential decision problems, as well. In *International Conference on Machine Learning*, pages 1757–1765. PMLR.
- Spaan, M. T. J. (2012). Partially Observable Markov Decision Processes. In Wiering, M. and van Otterlo, M., editors, *Reinforcement Learning*, volume 12, pages 387–414. Springer Berlin Heidelberg, Berlin, Heidelberg. Series Title: Adaptation, Learning, and Optimization.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958. Publisher: JMLR. org.
- TensorFlow (2022). TensorFlow.
- Tsividis, P. A., Loula, J., Burga, J., Foss, N., Campero, A., Pouncy, T., Gershman, S. J., and Tenenbaum, J. B. (2021). Human-Level Reinforcement Learning through Theory-Based Modeling, Exploration, and Planning.
- van der Wel, P. and van Steenbergen, H. (2018). Pupil dilation as an index of effort in cognitive control tasks: A review. *Psychon Bull Rev*, 25(6):2005–2015.
- van Opheusden, B. and Ma, W. J. (2019). Tasks for aligning human and machine planning. *Current Opinion in Behavioral Sciences*, 29:127–133.
- van Renswoude, D. R., Raijmakers, M. E., Koornneef, A., Johnson, S. P., Hunnius, S., and Visser, I. (2018). GazePath: An eye-tracking analysis tool that accounts for individual differences and data quality. *Behavior research methods*, 50(2):834–852. Publisher: Springer.
- Volkman, F. C. (1986). Human visual suppression. *Vision Research*, 26(9):1401–1416.
- Volkman, F. C., Riggs, L. A., and Moore, R. K. (1980). Eyeblinks and Visual Suppression. *Science*, 207(4433):900–902.
- Wang, R. E., Wu, S. A., Evans, J. A., Tenenbaum, J. B., Parkes, D. C., and Kleiman-Weiner, M. (2020). Too many cooks: Coordinating multi-agent collaboration through inverse planning.
- Wheaton, G. R. (1968). Development of a taxonomy of human performance: A review of classificatory systems relating to tasks and performance. Publisher: AMERICAN INSTITUTES FOR RESEARCH PITTSBURGH PA.
- Yang, S. C.-H., Lengyel, M., and Wolpert, D. M. (2016). Active sensing in the categorization of visual patterns. *Elife*, 5:e12215. Publisher: eLife Sciences Publications Limited.
- Zhang, R., Walshe, C., Liu, Z., Guan, L., Muller, K. S., Whritner, J. A., Zhang, L., Hayhoe, M. M., and Ballard, D. H. (2019). Atari-HEAD: Atari Human Eye-Tracking and Demonstration Dataset.
- Zhu, S., Lakshminarasimhan, K. J., Arfaei, N., and Angelaki, D. E. (2022). Eye movements reveal spatiotemporal dynamics of visually-informed planning in navigation. *eLife*, 11:e73097.

A. Appendix

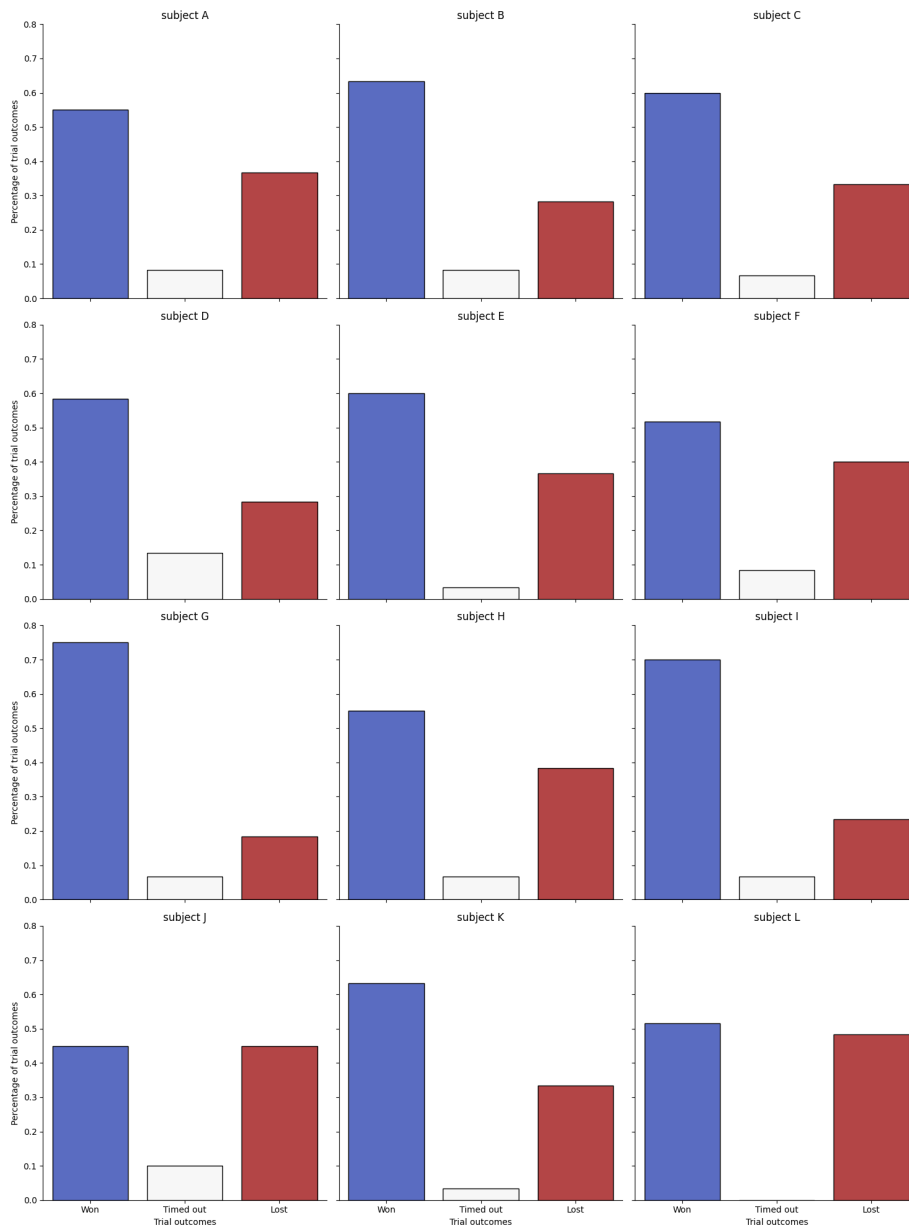


Figure A.1: Trial outcomes per subject in percent of all completed 60 trials (20 trials for each difficulty). For details on trial order and difficulty tuning, see section 3.