# Computational modelling of cognitive flexibility behaviour in a mouse model of vascular dementia

*Peiheng Lu*


4th Year Project Report
Computer Science
School of Informatics
University of Edinburgh

2023

# Abstract

This is a pilot study laying down the foundation for a larger-scale publishable research.

This study quantitatively investigates the effect of vascular dementia (VaD) on cognitive flexibility. The study used a dynamic foraging task on a mouse model with progressive pericyte loss, $Pdgf\beta^{F7/F7}$, which exhibits alternations commonly found in VaD. Prior to disease onset, F7/F7 animals were trained on the task, and quantitative methods including raw performance metrics and reinforcement learning modeling were used to track disease progression and understand cognitive alterations in VaD.

Modelling did not produce satisfactory result, the parameters of candidate models did not seem to provide much insight into the thought process of disease-type animals, and the data collected so far were too scarce for concrete conclusions to be drawn. However, the experiment design and the custom setup were proven to be viable, and are ready for deployment in larger scale. Limited evidence were also shown in support of the hypothesis that animals can show impaired cognition in this iteration of dynamic foraging task.

# Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

The animal procedures received written permission from BVS, and is conducted under project license (PPL) PP8564759. I was granted permission to conduct animal work, my personal license (PIL) is I12862471.

# Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Peiheng Lu*)

# Acknowledgements

Thank you to Dr. Gan Jian, who kindly sponsored the expenses for this project, provided the location, animals, helped me acquired the license needed for animal work, and give me important suggestions on where the project is heading.

Thank you to Dr. Wei Xu for coming up with the idea of the project, helping me get started with electronic engineering and signal processing, and given some import suggestions on the writing.

Thank you to Mingshan for collecting a lot of the animal behaviour data, this study would not have been possible without your support.

Thank you to Filippo for patiently introducing me to the process of simulation optimization pipeline.

Thank you to Prof Peggy for your suggestions during our meetings and your quick but incredibly detailed email replies, they have all been very informative.

I'm forever in debt for the amount of time and effort that you have put into this project, helping me build everything from ground up. The result may not have been as ideal as we originally envisioned, but I still learned a lot from this one-year long adventure. I wish you all the very best in your future journey, and I will do what I can to come to your assitance if it's ever needed.

# Table of Contents

# Chapter 1

# Introduction

## 1.1 Vascular Dementia and *F7/F7* Animals

Vascular dementia (VaD) ranks as the second most prevalent form of dementia, and arises from either global or focal effects of vascular disease. The disorder is characterized by a range of neurocognitive deficits, including behavioral symptoms and locomotor abnormalities such as dysarthria, autonomic dysfunction, and a Parkinsonian-like gait disorder (Kalaria 2018). The notion of VCID (vascular dementia and/or vascular cognitive impairment and/or vascular contributions to dementia) has emerged as a significant priority within the field of neurology, following its identification as such at the Alzheimer's Disease-Related Dementia (ADRD) Summit in 2013. Since then, there has been a global emphasis on identifying effective treatments for the condition.

One of the hallmarks of VaD is poor decision-making leading to inflexible and repetitive behaviors. For instance, if a person's usual route to work is closed down for maintenance, an individual with healthy cognition would be able to find an alternative route and still arrive on time. However, this task would be challenging for patients with cognitive flexibility impairment.

Research has revealed that various vascular structures play a role in causing cognitive impairment in dementia patients. Among these, the pericyte has attracted considerable amount of interest from researchers investigating dementia (Ding et al. 2020, Shi et al. 2020).

Pericytes are specialized mural cells that reside within the basement membrane and encircle microvessels, as illustrated in Figure 2.1. These cells play various essential roles, such as regulating the neurovascular system by ensuring vascular stability, vessel formation, and cerebral blood flow. Additionally, they aid in trafficking inflammatory cells and clearing toxic waste products from the brain. Of utmost significance, pericytes are responsible for developing and maintaining the integrity of the blood-brain barrier (BBB) (Uemura et al. 2020). Detachment of pericytes from microvessels can lead to a breakdown of the blood-brain barrier, resulting in white matter diffusion and impaired cognition by hindering communication between cortical and subcortical areas (Zhang et al. 2019).

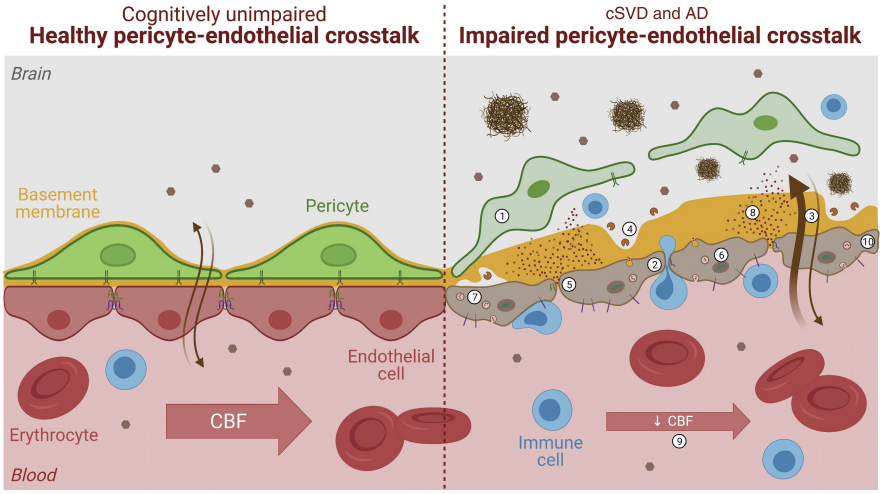



Figure 1.1: Healthy and Impaired Pericyte. Illustration From Procter, Williams, and Montagne 2021

Recent advances in genetics have enabled the creation of a mouse model that undergoes gradual and controlled pericyte loss. One such model is $Pdgfrb^{F7/F7}$ (referred to as *F7/F7* animals for brevity), which is a"pericyte-deficient mouse strain carrying seven point mutations in the platelet-derived growth factor receptor β ($Pdgfr\beta$) gene. These mutations disrupt $Pdgfr\beta$ signaling in vascular mural cells", ultimately causing pericyte loss (Montagne et al. 2018). The $F7/F7$ animals' pericyte loss can result in blood-brain barrier breakdown and microvascular alterations that resemble changes observed in vascular dementia(Sweeney, Ayyadurai, and Zlokovic 2018, Kumar et al. 2021), making it a suitable candidate for this study.

## 1.2 Dynamic Foraging

Natural environments are characterised by uncertainty in both the sources and timing of rewards. Humans and other animals can adapt their foraging behaviour to those of the environment. Specifically, animals distribute their time among foraging sites in proportion to the relative abundance of resources at each site. (Kahneman and Tversky 2013). The matching tasks are designed to simulate a foraging environment and test if the animals' behaviour is aligned with the matching law ($\frac{I_k}{\sum I} = \frac{C_k}{\sum C}$, where $I_k$ and $C_k$ are the total income earned and total choices on option k, respectively) formulated by Herrnstein (Herrnstein 1961). In a dynamic version of the matching task, the probability of rewards (abundance of food) is not stationary. As the task progress, the reward probability associated with each choice changes.

This is formally called a reversal learning task in Psychology, and "is known as a 'pre-eminent test of cognitive flexibility', not least because it is observed ubiquitously and is also easily quantified in different species" (Birrell and Brown 2018). The dynamic foraging task is a reversal learning task that rewards with food or water, and thus has the ability to gauge the participants' cognitive flexibility, an aspect particularly affected by VaD.

### 1.2.1 Dynamic Foraging Task in This Study

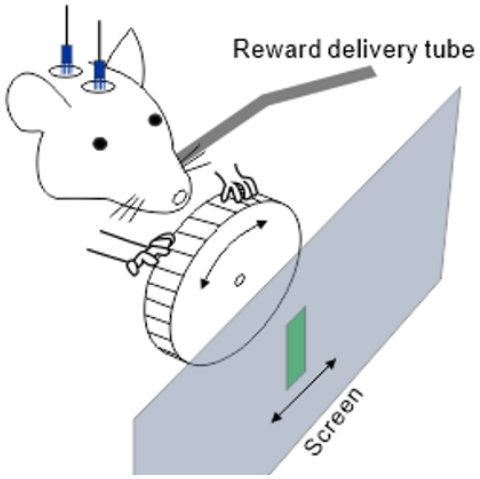



Figure 1.2: Illustration of Animals Doing the Task, Provided by Dr. Wei Xu

To better understand this task, Figure 1.2 presents a simple depiction of the version used in this project.

In essence, this iteration of the dynamic foraging task is a non-stationary 2-arm bandit task. In each trial, the animal chooses between turning left or right and receives a corresponding reward. The animal makes decisions by manipulating a steering wheel in front of them, whose movement is visualized by a green cursor on the screen. The animals must rotate the wheel past a specific threshold to register their selection, and this threshold varies across different training stages.

Each side of the task is linked to a reward probability, and a 5% sucrose solution is dispensed from the reward delivery tube as a reward. The amount of reward given per trial is also variable in each stage to ensure that the animals receive sufficient water intake from the task itself, reducing the need for supplemental water and preserving the animals' motivation.

Cognitive flexibility is reflected in this task as the animals' ability to adapt to a probability switch. A probability switch occurs after the animals are able to consistently choose the advantageous side. The next subsection will describe the rule of the switch test, along with specific terms that describe various aspects of the task.

### 1.2.2 Describing a Session of the Task

Certain terminologies will be used in this article to describe a session of the task, and Figure 1.3 shows an example of how a session's data is visualized in this experiment.

Vertical ticks represent the animal's response in each trial, with ticks on top indicating choices made towards the right side and the ones on the bottom indicating left choices. A blue tick signifies that the animal was rewarded in that trial, while a red tick implies no reward was given.

Trials where the animals did not make a choice are referred to as **NaN trials**, which are indicated by a black tick at both the top and bottom of the figure.

The trial where the reward probability changes is referred to as a **switch**, while the trials between two switches constitute a **block**. Each block has a random minimum length within the range of 50-80 trials, and after the current block's number of trials has surpassed the minimum block length, a switch test is conducted every 20 trials to determine whether a probability switch should occur. If 15 out of the past 20 trials involve choices towards the advantageous side, a probability switch takes place. These

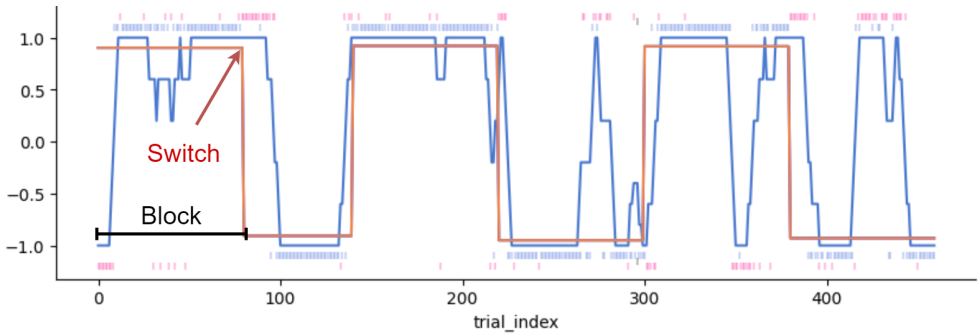



Figure 1.3: Example Visualization of a Session

switch test parameters have been derived from past experience in a similar task in Dr. Jian Gan's lab, where they had achieved good training results.

Note that the start and end of the session do not count towards the number of switches in a session. The orange line denotes the reward probability of the advantageous side, while the blue line indicates the animal's choices, averaged using a moving window of size 5 and centred at the current trial.

## 1.3 Computational Neuroscience, Computational Psychiatry and Reinforcement Learning

Computational neuroscience is an interdisciplinary field that aims to develop mathematical and computational models of neural activity to better understand how the brain processes and encodes information. These models can simulate neural circuits and predict how they will respond to different stimuli and tasks. With these models, researchers can test hypotheses about the neural mechanisms underlying behaviour and make predictions about how the brain may respond to different types of stimuli or cognitive challenges (Abbott and Dayan 2001).

Computational psychiatry is a relatively new field that bridges the gap between computational neuroscience and psychiatric disorders, aiming to gain a deeper understanding of mental illness. By applying computational modelling and theoretical approaches, it seeks to understand how and why the nervous system may process information in dysregulated ways. Specifically, computational psychiatry seeks to answer three key questions in the study of psychopathology (Series 2020):

1. What are the main biological components involved in psychopathology and what is their mathematical relationship?
2. How their dysfunctions lead to mental illness
3. What is the aetiology of the disease?

In this project, the focus is on gaining a better understanding of the first two questions, as the aetiology of vascular dementia in the *F7/F7* mouse model is already known.

Various methods are currently being applied in the field of computational psychiatry,

including deep learning models, Bayesian statistics, and reinforcement learning. These methods can also be combined to create more complex models, such as deep reinforcement learning. As a pilot study, simple reinforcement learning models, such as the Rescorla Wagner model, were implemented to explore whether these rudimentary models can already provide useful insights into animal behaviour.

Reinforcement learning is a type of machine learning that involves learning through interaction with an environment by taking different actions and experiencing various successes and failures to maximize rewards. It is considered the third paradigm of machine learning, along with unsupervised and supervised learning(Hammoudeh 2018).

An enormous amount of publications have been discussing the parallels between the neuroscience of learning and decision-making to reinforcement learning (Ludvig, Bellemare, and Pearson 2011, Glimcher 2011). One of the most remarkable contacts is the connection between reinforcement learning and the involvement of dopamine in reward processing in the brain of mammals. Dopamine appears to convey temporal difference errors to brain structures where learning and decision-making take place. This parallel is expressed by the reward prediction error hypothesis of dopamine neuron activity, a hypothesis that resulted from the convergence of computational reinforcement learning and the results of neuroscience experiments. The predominant concept is that the phasic dopamine activity signals actual versus expected reward values, or a reward "prediction error". This prediction error, in turn, is thought to play a key role in rewarded learning and has gained widespread use in temporal difference models of learning that are driven by reinforcing rewards (Sutton and Barto 2018, Gazzaniga, Ivry, and Mangun 2019).

The evidence of reinforcement learning in decision-making processes in biological systems has inspired this study to use reinforcement learning models as the primary method of investigation.

## 1.4 Hypothesis, Project Structure and Project aim

There have been several studies conducted on reversal learning tasks, including modeling these tasks on healthy participants using reinforcement learning models like the Rescolar Wagner models (Bari et al. 2019), studying participants with cognitive impairment (especially gambling disorder) using overall performance metrics in reversal learning tasks (Perandrés-Gómez et al. 2021, Jara-Rizzo et al. 2020), and modeling these tasks on participants who have already been diagnosed with cognitive impairment, again using the R-W model (Peterson et al. 2009). However, there are few studies that track the progression of cognitive impairment of participants using rigorous model comparison and fitting processes. This study aims to lay the foundation for a bigger research project that fills this gap.

Using the F7/F7 animals, the current project is looking into the following hypotheses:

1. Changes in cognitive flexibility can be reflected in this task as difficulty during training or declined performance for a trained animal;

2. Changes in cognitive flexibility can be reflected through changes in fitted parameters in certain models or through the type of models that best describe the animals'

behaviour in a version of the dynamic foraging tasks.

To test these hypotheses, the following steps are needed:

1. Designing the Experiment

   A well-designed experiment should aim to minimize the variability of control variables among participants, such as age and health conditions. This requires a quantitative definition of parameters including session length and task difficulty, at different stages of the animal's training process. Additionally, a benchmarking method should be established to determine when an animal is sufficiently trained and ready for data collection.

2. Constructing a Dynamic Foraging Tasks Setup

   It is also necessary to construct the physical apparatus utilized for data collection of animal behaviour. This involves developing the software that controls the behavioural tasks and constructing the hardware using tools including 3D printing.

3. Verifying the Task Setup

   The setup's viability for animal training and data collection should be tested before any further analysis is conducted.

   The more obvious test is to qualitatively compare the animals' performance. Are they actually learning the task? And are they passing the benchmark in a reasonable amount of time? If the sessions look reasonable qualitatively, some quantitative analysis of the training data can be run after the training is finished for more concrete results. Specifically, two conditions should be true here: (1) The performance should have a noticeable upward trend during the training of the animals. (2) The speed of adjusting to the new advantageous side after a switch should be increased during training.

   Both of the above conditions are true for the Pheonosys rig, which has been proven to be viable, thus a good result here should increase the credibility of the study, while a poor result with either of these conditions being false can be a great indicator of possible mistakes in the setup.

4. Modelling the Task

   Initially, a set of candidate models should be proposed, ideally models that employ distinct assumptions regarding the cognitive processes employed by animals when performing the task, such as those that describe associative or rule-based thinking.

   Subsequently, a sequence of model selection tasks must be carried out to eliminate inadequate models. Once these tasks are completed, the primary hypothesis of the study can be investigated, and the chosen models are fitted onto actual animal data to compare their parameters.

In short, the project aims to use a progressive-pericyte-deficient mouse line and a custom dynamic foraging task setup to quantitatively explore the impact of pericyte loss on learning and cognitive flexibility, a key symptom of cognitive impairment observed in Vascular Dementia.

# Chapter 2

# Method

This chapter presents the experimental design implemented in this project, in addition to providing specific information pertaining to the construction of the experimental apparatus and quantification of task parameters.

## 2.1 Experiment Design

Four $Pdgfrb^{F7/F7}$ animals were employed in this experiment, alongside four healthy control group animals from the same strain. As depicted in Figure 2.1, *F7/F7* animals exhibit early and progressive pericyte loss. The coverage of pericytes for *F7/F7* animals is consistently lower, but noticeable changes in cognitive ability only become apparent after they reach 12-16 weeks of age. This sets a deadline for all training to be completed before the animal reaches 16 weeks old (Montagne et al. 2018). If an animal cannot be trained within that timeframe, it was excluded from the experiment. The experiment commenced when the animal was aged between 45 and 50 days, allowing for the closest possible age matching between groups. Nonetheless, past experience in the task has demonstrated that individual performances can still vary significantly, even for normal mice from the same strain at the same age. Some animals can acquire a good grasp of the task in fewer than 30 training sessions, while others may require 60.

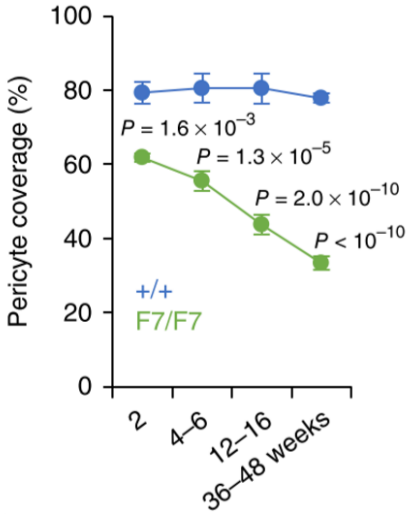



Figure 2.1: percentage of pericyte loss for *F7/F7* animals. The blue line represents the pericyte coverage of healthy control group, and green line shows the coverage for mutated *F7/F7* animals. Illustration from Montagne et al. 2018.

To effectively control individual variability, data for modelling were collected after the animals had learned the task sufficiently to pass the same benchmark. The benchmarking details are introduced in subsection 2.1.2 and section 2.3. This approach ensures that the animals,

both the disease type and the control group, possess a similar level of cognition as measured by this task when data collection for modelling begins.

All participating animals underwent three major stages: preparation, training, and data collection. The experiment pipeline for an animal is shown in Figure 2.2 below and is elaborated upon in the subsequent subsections.

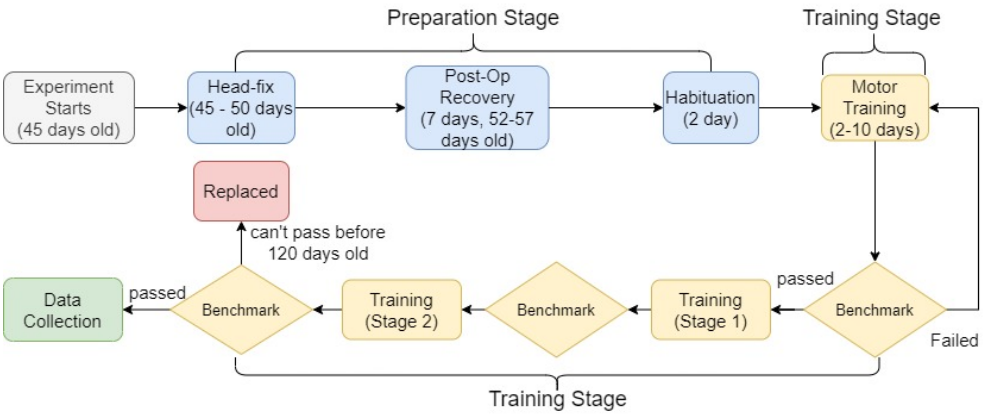



Figure 2.2: Animal Experiment Pipeline

### 2.1.1 Preparation

Following an inspection by a veterinarian, the animals underwent surgeries for head-fix implantation. Animals were then given seven days to recover from the operation, during which they also began habituating to the laboratory environment. After their body weight returned to pre-operation levels, water restriction commenced to motivate the animals in the behavioural task. The body weight at the start of water restriction was recorded as the initial body weight. To maintain the animals' health, a minimum of 40 ml/kg/day of fluid was provided. If an animal's body weight ever fell below 80% of the initial body weight during any part of the experiment, water restriction ceased for at least three days and would not resume until its weight qualified again (Barkus et al. 2022).

An additional two days of habituation followed the initiation of water restriction, during which the animals needed to acquire some " "survival basics". Since the water bottle was removed, it was necessary to ensure that they could learn to drink from a small glass container used to deliver extra water. The animals were also headfixed on the contraption for acclimation, helping them become more relaxed on the rig before the actual training began (Paré and Glavin 1986). When on the rig, the animals needed to learn to drink from the reward delivery system, which typically required some human assistance, but all animals were able to learn to drink from the tube quickly. It was crucial not to keep the animal restrained for an extended period during this stage, as prolonged head-fixing without behavioural tasks could lead to excessive stress, negatively impacting the animals' performance in the task (Stuart et al. 2013).

### 2.1.2 Training

The training of the mice involved three phases, with progression between each phase determined using benchmarks whose specific thresholds are introduced in section 2.3.

Training commenced with 2-5 days of **motor trainin**, during which the participants needed to understand that turning the wheels could earn them rewards. The animal learns to use a contraption by trying random actions and discovering the desired action that results in a reward (Thorndike 1898). The head-fix limited their possible actions on the rig, making it more likely for them to turn the wheels. Since an accurate record of the animals' choices was not required at this stage, the threshold for a choice to be registered was only three-fifths of what was needed for other stages. This made it easier for the animals to trigger the reward delivery system and"stumble upon" the connection that turning wheels could earn them rewards. This version of the task provided a 90% reward probability on both sides. A 100% reward probability was avoided to help prevent animals from developing a strong side bias during this period. According to past experience, having trials where they do not receive a reward encourages them to explore by turning the wheel to different sides. The animals were benchmarked daily after starting the motor training stage and would progress to the next stage after passing the test.

Following their success in the motor training benchmark, the animals entered the task training stage. They now needed to understand that the two directions of wheel turning could result in different reward probabilities, and that these probabilities could change during a session.

The first part of the task training stage, called **training_1**, featured a probability difference between the two sides greater than 80%. The disadvantaged side provided a 5-10% chance of reward, compared with 90-95% on the opposite side. The reward probabilities of the two sides were complementary (summing to 1) to facilitate cleaner mathematical processes during analysis.

To obtain a better gauge of an animal's mental flexibility, a task should neither be too complex nor too simple. Normal animals are likely to perform as poorly as animals with deficits if the task is overly challenging. Conversely, disease-type animals can perform well for an extended period after disease onset if the task is too easy (Montagne et al. 2018). Consequently, a more challenging **training_2** stage followed if the animal passed the benchmark for training_1. The animal would now need to handle as little as 70% probability difference,the same range of difficulty during the data collection phase. The disadvantaged side now offered a 10-15% reward probability, with the advantaged side probability remaining complementary. The determination of these probability differences used in training_1 and 2 is introduced in subsection 2.3.3. Data collection started as soon as the animals passed the final benchmark for the training_2 stage.

### 2.1.3 Data Collection

After the animal managed to finish all training stages, the collection of data used for the model fitting and comparison can commence.

For better comparison across the animals, although the length of each block still depended on the animals' performance, the probability of reward for each block was controlled by a random seed. The random seed was used to produce the probability for the advantageous side, sampling from a uniform distribution in the range [0.85, 0.9] controlled by the seed. The animals worked through the same set of seeds $[0, 90)$ so that their probability for each block (that they can go through) was the same, and thus their performance can be compared on tasks of similar difficulty.

## 2.2 Dynamic Foraging Task Setup

### 2.2.1 Commercial Pheonosys Rig

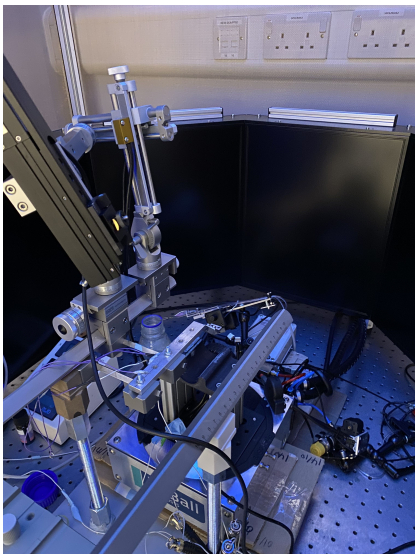


Figure 2.3: Pheonosys Rig

The dynamic foraging rig used in Dr. Gan Jian's lab is powered by computers custom-made by PhenoSys (hereafter referred to as the PhenoSys rig, shown in Figure 2.3) and is primarily suitable for electrophysiology recording. However, it offers less flexibility on the behavioural side of the task. Since fine-grain control over parameters, such as session length, reward probabilities, and the amount of reward delivered, is required, which is not offered by the closed-source PhenoSys rig, an open-source version of the dynamic foraging rig was built. This also allowed the opportunity to automate as many processes as possible through communication with the Azure remote database.

### 2.2.2 Hardware

Figure 2.4 demonstrates an overhead view of the setup used in this study. Instead of a high-performance and heavily modified computer used in the PhenoSys rig, a Raspberry Pi 4 was used to support this task, which significantly reduced the budget. The built-in GPIO ports of Raspberry Pi also saved the trouble and expenses of modifying a PC to transmit signals to and from various components of the tasks. The hardware components used in this experiment and their connection details are described below.

- **A** is the raspberry pi controlling the experiment. It's connected to two monitors. Monitor **M1** shows the animal and human operator the green cursor turned by the wheel **W**, while monitor **M2** (not available to the animal) provides some performance tracking information to give the operator a visual representation of the animals' choices, similar to the one shown in Figure 1.3.
- **B** is the device head-fixing the mice during the task, it was designed independently to be more suitable for 3D printing. The white mouse holder **C** was 3D printed

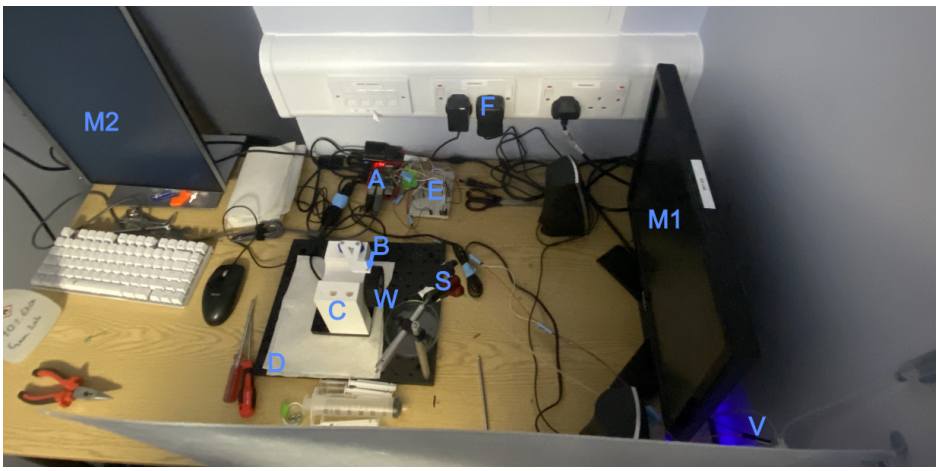



Figure 2.4: Simplified Dynamic Foraging Task Setup

using the model from UCL. The black base **D** where **C** is mounted on was purchased from RS according to the UCL specification of the task.

- The wheel **W** is connected to a 1024-tick rotary encoder mounted inside **C**. The rotary encoder records the movement of wheels as ticks and sends them to the raspberry pi, where the ticks are decoded into angular movement.
- Breadboard **E** has a simple circuit converting a 5v signal from the rotary encoder into 3.3V, the recommended voltage for the GPIO port. A 9v power supply **F** supplies the extra current needed to open the valve **V** and delivers the reward to the animals since the GPIO ports are not capable of providing enough current to drive any sort of motor. GPIO ports send signals to the transistors connecting the power supply and the valve to control the opening and closing of the valve. And finally, the water is delivered to the animal through sprout **S**

There is also a semi transparent film separating the animals and human operators to reduce the stress and effect of disturbance from the environment. At the same time, it blocks out some of the light for the animal, helping them stay more active as mice are nocturnal.

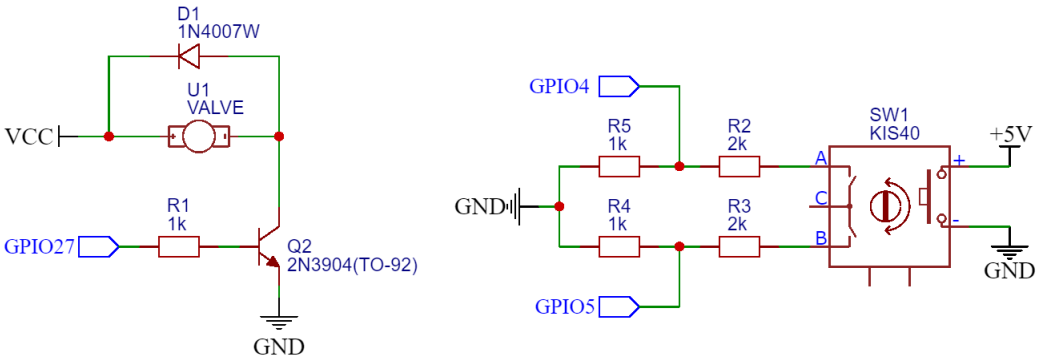



Figure 2.5: Schematics

Figure 2.5 shows the schematic of the electronic hardware components. All devices are

connected to Pi's common ground, and the 5V power supply for the rotary encoder is drawn from pi while a 9V power adaptor supplies the current needed by the valve. The two voltage splitters composed of R2, R5, as well as R3 to R4 are used to lower the output voltage of the rotary encoder from 5V to 3.3V. R4 and R5 also act as pull-down resistors to keep the inputs to Pi at 0 when no pulses are produced by A and B from the rotary encoder. Diode D1 is used to protect the GPIO ports from back EMF generated when the motor inside of the valve is shut down (Martin 2020).

### 2.2.3 Software

The software was mostly written from scratch by myself and is available at my Github repo. section B.1 walks through the important packages in this repository.

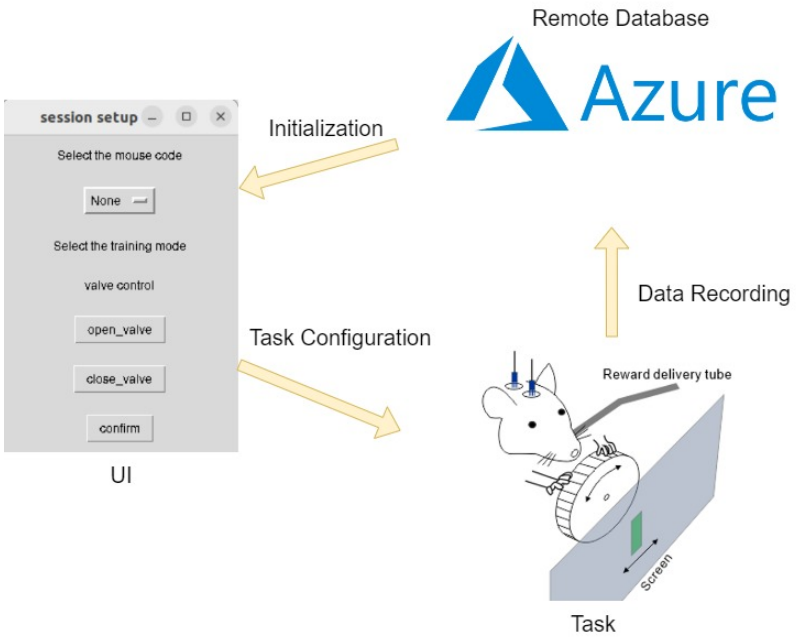



Figure 2.6: Component Interaction

Figure 2.6 demonstrates how different software components work together to power the task.

When the task starts, the program retrieves the list of animals currently doing the task. The GUI pops up and prompts the user to select the participant for the current session from the retrieved animal list, and maybe open the reward delivery valve when filling the tube with sucrose solution during the first session of the day. This is the only human input needed, the program then retrieves the information on animals doing the task from the database and installs the correct training profile. Then the task starts using information from the GUI and the database.

After the training session is over, the performance of each session is benchmarked, and any possible changes in training stages are recorded and uploaded to the database along

with session description data like choices, trial rewards, and reward probability of the two sides for each trial.

## 2.3 Determination of Task Parameters and Benchmarking Thresholds

Since this experiment is based on the task conducted on the PhenoSys rig, I can use our large amount of previously collected data to determine the most appropriate thresholds for assessing progression between the stages described in section 2.1. These data were also employed to qualitatively identify suitable values for session lengths, task difficulties and benchmark thresholds at each stage of the experiment.

The task in the PhenoSys rig is divided into different versions with varying difficulty levels. Most versions have a corresponding 'B' version (e.g., V21 and V21B). The base version has its first advantageous side starting from the left, while the "B" version starts from the right side. The most relevant versions for this study are:

- **V10**: Equal probability on both sides (85%). This is the first task every mouse is trained on, corresponding to the motor training stage.
- **V21 / V21B**: 5% reward probability on the disadvantaged side, against the 80 - 90% on the good side. This is the simplest version of the tasks where the reward probability differs for the two sides. Animals usually progress from V10 /V10C to this version. Naturally, it corresponds to the training_1 stage in our experiment.
- **V20 / V20B**: 15% reward probability on the disadvantaged side, against 80 - 90% on the good side. This is a frequently used session after the animal can perform adequately well in the V21 task, matching the training_2 stage

The overall statistics of all sessions are saved in the *"session_info.csv"* file. Some sessions are used for testing purposes or are interrupted by unforeseen factors, resulting in abnormally short session lengths. These sessions are removed by only logging the statistics of sessions with lengths longer than 100 trials.

### 2.3.1 Quantification of Task Performance

When benchmarking an animal, only the overall session performance is considered. Therefore, finding an appropriate quantification for task performance is crucial. Performance is measured differently for various task versions to quantify the ideal behaviour of a trained animal in the task. Note that for all performance measurements, higher values are better.

For the V10 (motor_training) version, session performance is evaluated using $1 - \textit{percentage of NaN trials}$. As animals become more familiar with the task equipment, they will gain confidence in turning the wheels, resulting in a lower percentage of trials where they fail to make a choice in time.

The V21, V20 (training_1, training_2), and data-collection versions have slightly more complex performance measurements. The first and most straightforward component is

the percentage of trials where the animals chose the side with a higher reward probability (the "**advantageous percentage**").

The number of block switches in a session is used as another part of the performance metric for training_1 and training_2, as a good session where the animals are following the advantageous side will have more switches according to the block-switching rule mentioned in subsection 1.2.2. To normalise the number of switches to the same range as the advantageous percentage ([0, 1]), the switch number is multiplied by 0.1. According to the switching rule, a maximum of 8 switches can be accomplished in a session with 450 trials, suggesting that the switch number should be multiplied by 0.125. However, some qualitative tests were conducted to see which value is producing the cleanest split between good and bad sessions in the range [0, 0.5] with a step size of 0.05, and 0.1 gives a cleaner split than 0.125. As a result, 0.1 was used as the weight here. Therefore, the performance measurement for the V21 and V20 versions can be expressed as:

$$Performance = Advantageous\ percentage + 0.1 * Number\ of\ switches$$

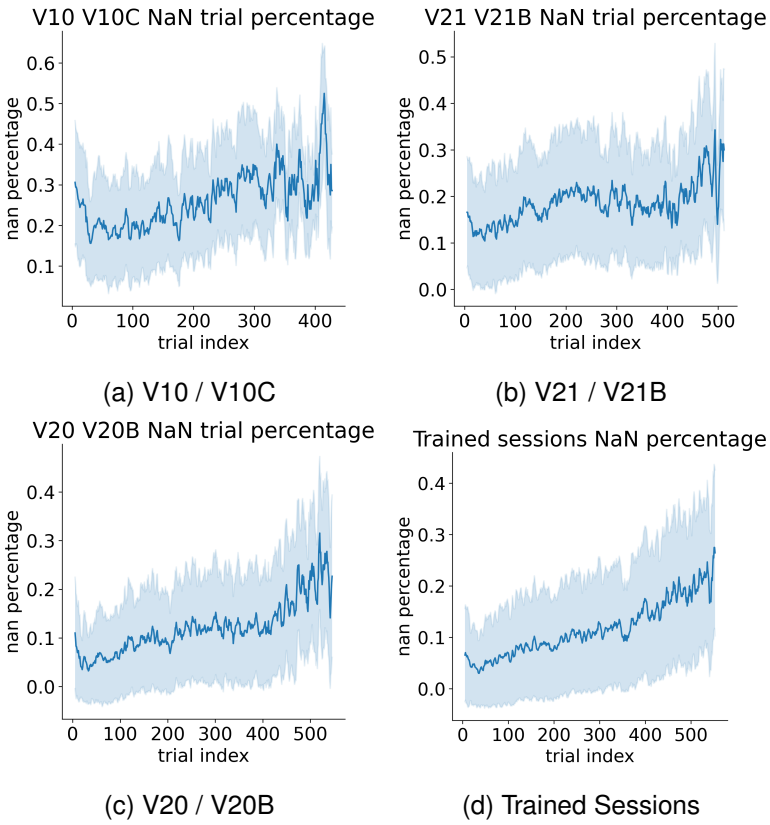
### 2.3.2 Determination of Session Length

Determining the optimal length of each version of the training sessions is crucial to ensure that animals are not stressed by remaining on the rig longer than necessary. Setting a session length limit also contributes to collecting higher-quality data, as fatigued animals are typically less engaged in the task.

A straightforward metric to assess an animal's exhaustion level is the percentage of NaN trials (trials in which animals fail to make a choice before the timeout). The percentage of NaN trials was calculated within a moving window of 5 trials, and the sessions were subsequently grouped by task version. The NaN percentage was averaged for each group using a modified version of "tolerant mean" (a function capable of calculating the mean of arrays with varying lengths without padding them to the same size). Trials with indices surpassing the 90th percentile of all sessions' lengths in their respective groups were clipped, excluding trial indices attainable by only 10% of the sessions and aiding the tolerant mean in producing more meaningful results. The results for V10 / V10C (68 sessions), V21 / V21B (146 sessions), and V20 / V20B (149 sessions) trials are plotted in Figure 2.9, with an error bar $\pm\frac{\sigma}{2}$.

The V10 NaN percentage displays an upward trend as the trial index increases and exhibits a relatively higher NaN percentage compared to other versions, which is expected since animals often begin sessions almost entirely composed of NaN trials. A trial limit of **300** was chosen for the motor training stage, as the NaN percentage of 0.3, a threshold defined for having "too many NaN trials", was reached around that index.

The NaN percentage of V21 and V20 trials, illustrated in Figure 2.7b and Figure 2.7c respectively, exhibit distinct elbow shapes, facilitating the identification of a reasonable stopping point. Although the elbow occurs around 400-450 trials for both versions, the V21 task's NaN percentage is substantially higher than that of V20. Consequently, the lower end of **400** trials is employed as the session length for the training_1 stage, while the upper end of **450** trials is utilized for training_2.




(a) V10 / V10C (b) V21 / V21B

(c) V20 / V20B (d) Trained Sessions

Figure 2.7: Average NaN Percentages

Regarding the determination of data collection session lengths, only PhenoSys sessions in which the animal is deemed "trained" are considered. Although no quantitative metric was established for the PhenoSys rig, a cursory examination of training data reveals that an animal can typically grasp the task after 30 training sessions (including motor training). The sessions for each animal were sorted by date, and the earliest 30 training sessions were filtered out. The remaining sessions were designated as the "**trained sessions**". The NaN percentage for the trained sessions is displayed in Figure 2.7d.

Interestingly, instead of the elbow-shaped plots generated by V20 and V21 versions, the increase in NaN percentage appears linear. Nonetheless, an acceleration in NaN percentage increase can be observed after approximately 450 trials, establishing the session length limit for data collection sessions at 450 trials, equal to the threshold for training_2.

For more details, C.1 contains some additional information on session length that are not as relevant to the understanding of this study.

### 2.3.3 Determination of Task Difficulty

As mentioned in subsection 2.1.2, a task should not be excessively difficult or simple. To identify the appropriate difficulty for each stage, the "**block performance**" was plotted against the probability difference for each block of the trained sessions" (*session index* $\geq$ 30), as shown in Figure 2.8. The block performance was calculated using the percentage

of trials in which animals chose the advantageous side during a block of a session. All versions employed by the PhenoSys rig (excluding those used for motor training) were included in this analysis to provide a more comprehensive range of difficulties, and the sessions were truncated using the 450-trial cut-off.Animals removed due to poor performances were also excluded from this analysis to minimise noise.

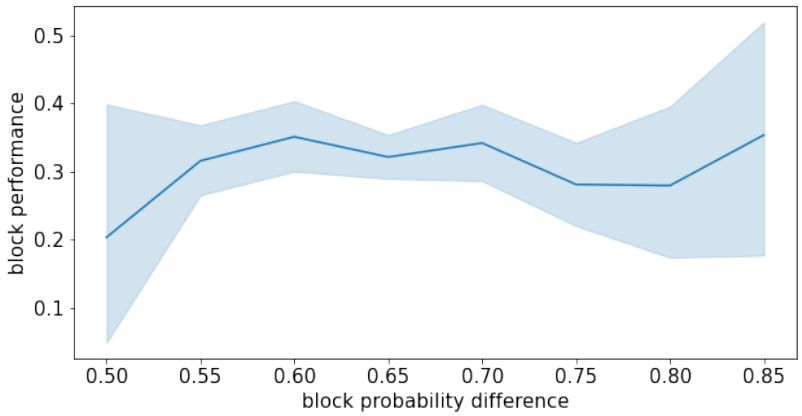



Figure 2.8: Task Difficulty vs Block Performance

Block performance does not seem to decline until the probability difference falls below 0.55, suggesting that even healthy animals struggle with sessions where the probability differs by less than 0.55. Both V21 and V20 versions of the task exhibit probability differences exceeding 0.55, so a similar range of difficulties to V21 and V20 is chosen for training_1 and training_2. Specifically, the probability difference for training_1 lies within the range of [0.8, 0.9], while the probability difference for training_2 is within the range of [0.7, 0.8].

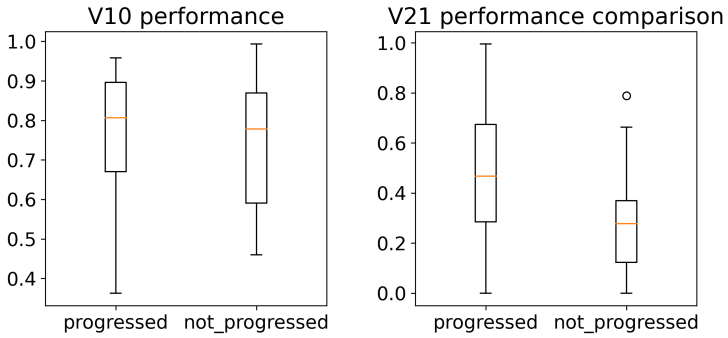
This also aids in determining the benchmark threshold in subsection 2.3.4, as existing sessions from the Phenosys rig can be utilised to quantify the behaviour of a well-trained animal.

### 2.3.4 Determination of Benchmarking Threshold

In our previous experiments, some animals were unable to progress any further than V21 as they either exhibited a strong bias towards a certain side or were simply uninterested in performing the task in numerous sessions. These were grouped as the "not progressed" animals. The comparison for "progressed" and "not progressed" animals can give us some important differentiation in how trained and not trained animals behave in the tasks.

- **Motor_training**

  The two groups exhibited no significant difference in their performance during the motor training stage (*two-sided t-test,* $p = .39$), with the progressed group demonstrating slightly higher mean performance. Consequently, the mean of the progressed group, **0.8**, was employed as the cut-off for benchmarking motor training.




(a) V10 performance (b) V21 performance

Figure 2.9: Performance Comparison of Progressed and not Progressed Animals

- **Training_1**

  Their V21 performances, however, differed significantly (two-sided t-test, $p <$ .0001). This allowed me to run binary classification to figure out the best way for splitting the "progressed" and "not progressed" sessions. Only the sessions with a session index greater than 10 were included. This constraint removed some of the sessions where both progressed and not progressed animals performed poorly due to lack of experience. The cutoff of 0.643 was found to be best separating the progressed and not-progressed sessions using logistic regression with liblinear solver and l1 regularization. Nevertheless, after visually inspecting the V21 sessions ranked by their performances (which can be found in the "info-extraction.ipynb" file within the "benchmark" subpackage), it was determined that 0.643 did not effectively filter out many poor sessions. Therefore, a more arbitrary method was applied, considering an animal to have passed this stage if its performance exceeded the 95th percentile of the "not progressed" V21 performances, which is 0.95.

  Qualitatively, **0.95** provided a much clearer cut-off between good and bad sessions and was used for benchmarking training_1 sessions.

- **Training_2**

  Benchmarking for training_2 is more challenging, as all animals that progress to V20 are well-trained, leaving no "not progressed" animals for comparison. The **0.95** threshold for training_1 was employed as the cut-off since the desired performance for these two task versions is the same, and sessions with a performance above 0.95 demonstrate the expected behaviour of a well-trained, healthy animal in this task.

# Chapter 3

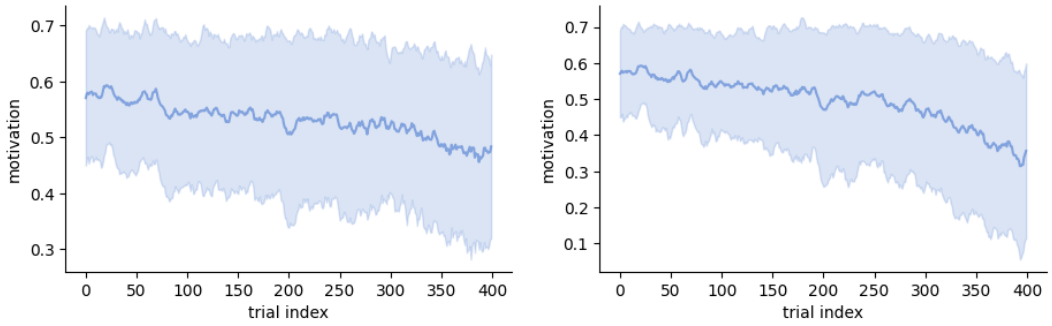
# Analysis and Result

Out of the 8 animals that participated in the experiment, the setup underwent some software and hardware changes during the training of PLED01-04, leading to the exclusion of their data from the analysis of this project.

Below is the complete list of variables used in the modelling process, the meaning of their values, and their unit(when applicable)

- **left_P**: the probability of the reward being delivered if the animal chooses left
- **righ_P**: the probability of the reward being delivered if the animal chooses right
- **choices**: the choice the animal or the model made in a trial, 0 means left, 1 means right, -1 for nan trials
- **rewarded**: whether the animal was rewarded or not in a trial, 0 means no reward, 1 means rewarded
- **reaction_time**: the amount of time (in seconds) between trial start signal is given and a choice is made by the animal. -1 if the animal didn't make a choice before a trial times out.
- **moving_speed**: the average speed of the animal moving the wheels during the reaction time (ticks/second). -1 if the animal didn't make a choice before trial times out.

To assess the level of motivation of mice during a session, their moving speed and reaction time were standardized to a range of 0-1 across all data collection sessions for each animal. The difference between their moving speed and reaction time was used as a simple indicator of motivation, which was further normalized to a range of [0, 1] across all data collection sessions for each individual animal. The NaN trials were assigned a motivation value of 0. The averaged motivations of all animals during data collection are visualized in Figure 3.1. To examine potential bias introduced by assigning a motivation value of 0 to NaN trials, the NaN trials were removed and the motivation estimates were recalculated, as shown in Figure 3.1a. Both figures show clear downward trends, indicating that animals became less motivated to perform the task and make switches during later stages of the session, consistent with our observations.




(a) Motivations Without NaN Trials (b) Motivations With NaN Trials

Figure 3.1: Motivations of Animals During Data Collection

As outlined in section 1.4, the analysis began with verifying the experimental setup using performance metrics and testing two conditions about animal performance during training. Variants of the win-stay lose-switch and Rescorla-Wagner models were proposed as potential candidates to model animal behaviour. These models were applied to the data using the simulation optimization pipeline described in section 3.2. Finally, the best-fitting models were selected to explain the animals' behaviour.

Another Github repository contains all the code written for the analysis in this chapter. section B.2 walks through the important components of this repository.

## 3.1 Pre-Modelling Analysis

### 3.1.1 Verification of the Performance Metric

I was introduced to another method to evaluate animals' performance during the later stages of the experiment, which is also quite intuitive and seems to produce a nice split between the good and bad sessions. It follows another way to explain what a "good" session is: if an animal is properly following the rules of the task, its choices should be matching the changes of the reward probability with a certain lag (phase shift). A cross correlation can adjust for the phase shift, and the maximum output of cross correlation should be where the two signals (reward probability of one side and the proportion of choices made to the same side) match the best. Ideally, a good session should produce a result close to 1.

Although the animals were already benchmarked with the performance metrics specified in subsection 2.3.1, it's not too late to compare these two metrics. Whether they correlate with each other or not should strengthen or weaken the results from this study.

To calculate the cross-correlated performance metric, the proportion of choices made to the left side as well as the reward probability assigned to the left side were both normalized to [0, 1], then the cross correlated signal of the two normalized arrays were calculated, vice versa for the right side. The mean of left and right cross correlated signals was used as the overall cross correlation, and the peak of the overall signal was used as the performance under the cross correlation metric.

| | PLED05 | PLED06 | PLED07 | PLED08 |
|---|---|---|---|---|
| pearson's R | 0.38 | 0.58 | 0.64 | -0.12 |
| MSE | 0.08 | 0.09 | 0.07 | 0.18 |

Table 3.1: Result of Performance Metric Comparison

For comparing these two metrics, the performances of all sessions from each mouse quantified by the two metrics were calculated. The result for PLED07 is shown in Figure 3.2 as an example, with the segmented blue line representing the performance using this study's performance metric, and the orange line representing the cross correlation metric. The correlation coefficient (Pearson's R) and the MSE between the normalised performances were calculated and recorded in Table 3.1. The visualization for other three animals can be found in section A.1.

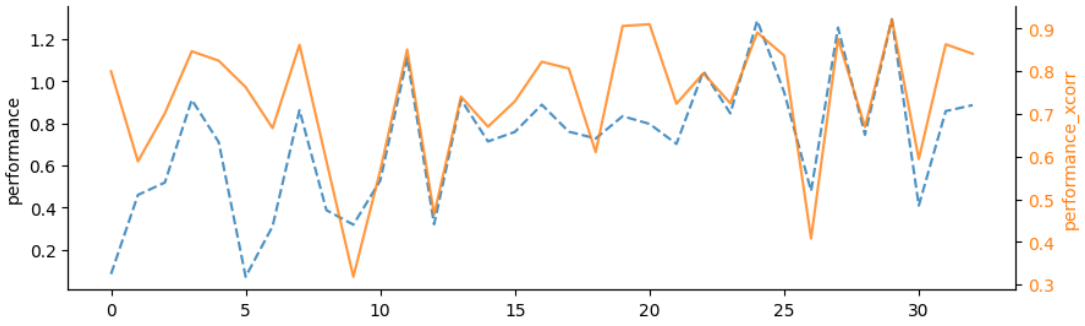
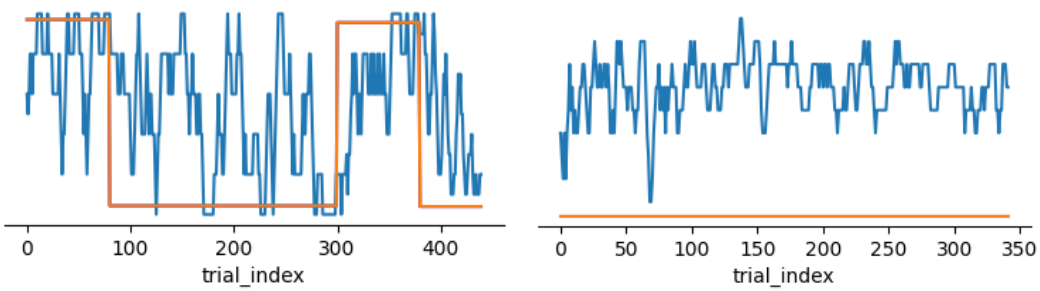



Figure 3.2: Example Result of Performance Metric Comparison

As you can see, for 3 out of the 4 animals, a moderate correlation is shown with a relatively small MSE, which is an encouraging sign. However, PLED08 seems to be an obvious outlier out of the three animals, who also showed the most significant difficulty during the training process.

To understand where the discrepancies between the two metrics came from, all the sessions where the normalised performances produced by these two metrics differ by more than 0.1 were plotted and inspected, and they were clearly divided into two groups.




(a) Great Difference With Messy Switches (b) Great Difference With Few Switches

Figure 3.3: Example of Sessions With Large Differences

Most of the sessions where the two metrics have major disagreement are either:

1. Animals manage to trigger the switch, but are not confident in their choices or prefer to explore, as shown in Figure 3.3a. The percentage of rewarded trials is not high, and the number of switches is not large enough to bump up the performance, producing a low-performance value.

   Personally, I prefer this study's metric's decision to assign relatively higher performance to these sessions, since animals are actively engaging in the task and following the rules of the task, although not perfectly.

2. Animals cannot trigger a single switch, as shown in Figure 3.3b.

   I also prefer this study's metric's decision to punish the animals' behaviour since they do not seem to understand what the task expects them to do.

As for the special case of PLED08, 23 out of 32 of its training sessions are sessions like Figure 3.3b, with zero or only one switch, and 7 out of 32 of its training sessions are sessions like Figure 3.3a. This caused large disagreement between the two metrics.

Interestingly, for PLED06 and PLED07, the two metrics appear to agree a lot more in the later stage of the training process. The performance array were ordered by date and split in half. Each half was evaluated using both of these metrics, and the Pearson's R as well as the MSE between the performances for each half were calculated, with the result shown in Table 3.2. Much stronger correlation and smaller MSE can be seen in the later stages of the training than the earlier stages, showing that the two metrics tend to agree significantly more on cleaner sessions like Figure 3.4, which is desirable.

| | PLED05 | PLED06 | PLED07 | PLED08 |
|---|---|---|---|---|
| first half pearson's R | 0.37 | 0.27 | 0.47 | -0.29 |
| second half pearson's R | 0.39 | 0.85 | 0.75 | 0.18 |
| first half MSE | 0.07 | 0.14 | 0.11 | 0.23 |
| second half MSE | 0.09 | 0.04 | 0.03 | 0.14 |

Table 3.2: Result of Split Performance Metric Comparison

In conclusion, the comparison with the cross-correlation metric showed that the metric used in this study is reasonable. However, if you prefer cross-correlation metric's decision on sessions like Figure 3.3a, then you should take the result from the rest of this analysis with a grain of salt.

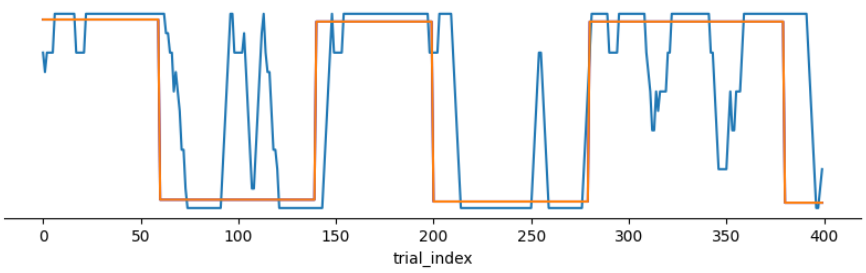



Figure 3.4: Example of Clean Session

### 3.1.2 Session-wise Performance Evaluation

After establishing the reliability of the performance metric used in this study, the first condition a working setup should satisfy is tested: animals' performance should exhibit an upward trend during each stage of training.

This was a simple analysis where the performance for each sessions during the training_1 stages of each animals were calculated. Linear regression was used to better illustrate the trend of the performance. The result is shown in Figure 3.5, where the dotted line shows the session performance smoothened over 5 sessions for better visibility, while the line with corresponding colour is the result of fitting a linear regression model onto the unsmoothened performance data for each mouse. A noticeable upward trend can be seen from all of the mice, even for PLED08, the only animal that failed to pass the final benchmark.

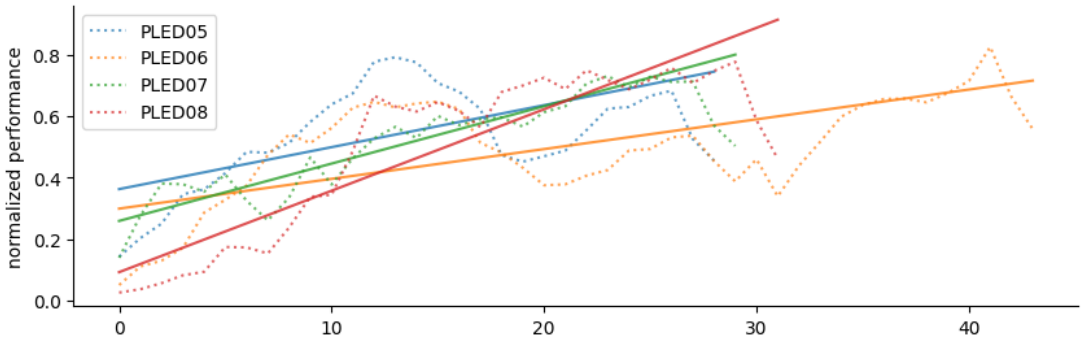
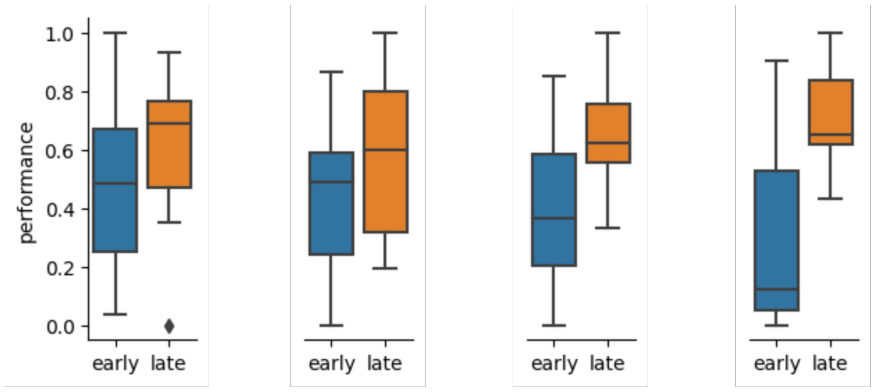



Figure 3.5: Session Performances During Training_1 Stage For All Animals

It is worth noting that the performances of PLED05 and PLED06 actually decreased after session number 10, possibly due to an interruption in training that lasted for around one month during the Christmas break. However, all the training data before the interruption were preserved in this analysis since the influence of training interruption is unclear, and it cannot be assumed that the animals started from a blank canvas after the training restarted.




a. PLED05 b. PLED06 c. PLED07 d. PLED08

Figure 3.6: Comparison Between Performances of Early and Late Sessions

To further confirm the result, the performances were ordered by session time, and split evenly into early and late sessions. The split performances were plotted with a boxplot, and the result is shown in Figure 3.6. For all four animals, the performances of later stages seem noticeably larger, and there is even less overlapping for PLED07 and PLED08, whose training was uninterrupted.

| | PLED05 | PLED06 | PLED07 | PLED08 |
|---|---|---|---|---|
| t statistics | -1.37 | -1.87 | -3.39 | -0.47 |
| p value | 0.09 | 0.03* | 0.00* | 0.00* |

Table 3.3: Result of t-test With “less” Hypothesis, * Means Significant

A t-test was also conducted with the hypothesis that the mean of the distribution underlying the early sessions is less than the mean of the distribution underlying the late sessions. The result of the t-test is recorded in Table 3.3. The result confirms our previous observation, with PLED05 and PLED06 confirming the hypothesis (rejecting the null hypothesis) with weaker 0.1 and 0.05 thresholds, respectively, possibly due to the interruption in training, while PLED07 and PLED08 confirmed the hypothesis with a stronger 0.01 threshold.

Similar analyses were not conducted on training_2 data since animals that can pass the benchmark for training_1 can pass the benchmark for training_2 in around four to five sessions. This was surprising as animals on the Phenosys Rig usually need to spend more time adapting to produce a good session after switching to the harder version. It could be that the 50 extra trials in the training_2 version give the animals more chances to make more switches, but the influence of that is limited since the minimum block size is 50, and thus the maximum advantage an animal can gain from the extra trials is 0.1.

This finding could be interesting for increasing training efficiency for further study in experiments like this. For example, patiently training the animal to greater proficiency in the easier version of the task may actually speed up the entire training process. It also suggests that a higher difficulty may be needed (for example, probability difference in the range of [0.6, 0.7]) for the training_2 stage to be meaningful.

### 3.1.3 Switch-wise Performance Evaluation

After confirming that animals’ training on this device looked normal session-wise, the second condition was tested, checking if the mice would react to changes in reward probability faster after substantial training.

To quantify this, all the sessions for each animal were again sorted by date and split into two even groups for comparison, one group for earlier sessions and one group for later ones. Then, all the switches in these sessions were extracted, each as an array of choices (20 trials before the switch and 20 trials after the switch). The result is plotted in Figure 3.7, where the “adv choice percentage” on the y-axis represents the percentage of choices made to the advantageous side, and the x-axis shows the trial index relative to the switch. The orange line shows the advantageous percentage of switches in the

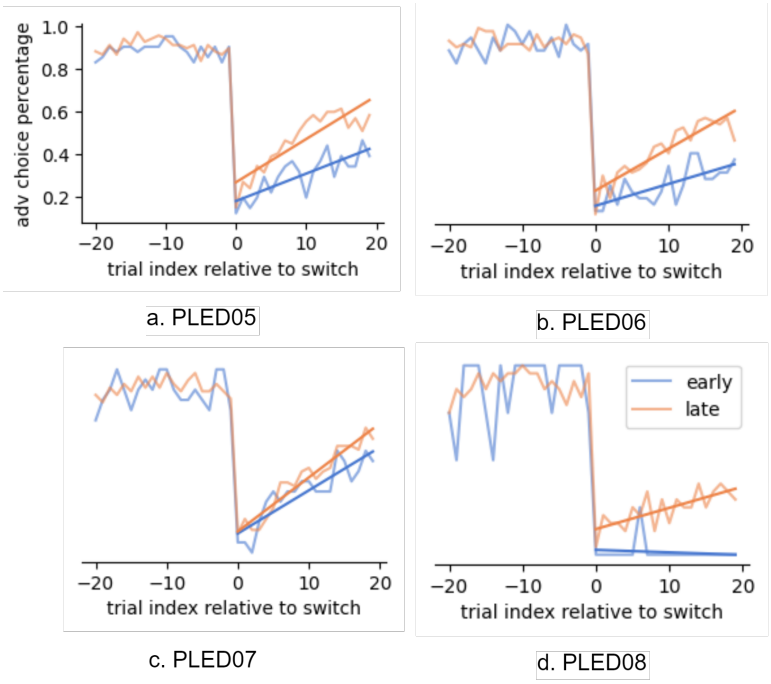



Figure 3.7: Averaged Switches For PLED05-08, legend in d. applies to all subfigures

"late" sessions, while the blue line shows the percentage for the "early" sessions. Linear regression was also applied for trials 0-20 (relative to switches), and the line segment with the matching colour shows the resulting linear equation.

As you can see, the slope of the fitted lines for the later sessions is noticeably larger than the slope for the early sessions in the training. This suggests that the animals were able to adapt to the changes in probability better after a number of training sessions, which is one of the most important goals of the training.

It is worth noting that PLED08 made very few switches, with only four switches in the earlier sessions and 24 switches in the later sessions, causing its line plot to appear more jiggly than others. Nonetheless, the same conclusion can still be drawn from its visualization.

## 3.2 Simulation Optimization Pipeline

A robust model comparison pipeline has been proposed to determine the most suitable model for describing animal behaviour (Wilson and Collins 2019). The simulation-based portion of the pipeline was implemented, which eliminates models with extraneous parameters or unexpected behaviour in the task. This three-step process includes simulation, parameter recovery, and model recovery, each of which will be discussed in detail in the following subsections.

Rescorla Wagner models is not a TD learning model that has a strong biological connection, as it provides point estimates for immediate reward instead of future cumulative reward (Mizunami, Terao, and Alvarez 2018). However, it is suitable for

non-stationary tasks like this. It was also shown that animals have internal representation for rewards expectation in 2-arm bandit tasks like this iteration of the dynamic foraging tasks, which aligns with the fundamental concept of R-W model (Bari et al. 2019). Therefore, the search for a well-fitting model began with the Rescorla-Wagner model.

In addition to the associative-based reinforcement learning models, rule-based Win-Stay-Lose-Switch models have also been a popular and successful choice in binary decision-making tasks (Gluck and Bower 2014, Denrell and Liu 2012). As a result, a WSLS model was also included in this study.

1. Win–stay, lose–switch

   This basic model serves as a baseline for the ability in describing animals' behaviour in the task. The stochastic WSLS model was used in this study, where the choice made during a trial depends on the result of the previous trial and the win-stay($p$) as well as the lose-switch($q$) probability. If the last trial was not rewarded, then the option not chosen in the last trial would be selected with a lose-shift probability. Conversely, if the last trial was rewarded, the option chosen in the last trial will be selected with the win-stay probability.

   The reward in this task is relatively frequent and consistent, thus a high win-stay probability of 0.99 was used as default, and the default for lose-shift probability was 0.2. No hard requirement was in place for the two probabilities to be complimentary.

2. Rescorla Wagner models

   In its full form, the R-W model updates the associated value for conditional stimulus(CS) A with:

   $$V_A = V_A + \alpha_A \cdot \alpha(\lambda_U - V_{AB})$$

   Where $\lambda_U$ is a variable that represents the maximum possible association strength for the CS. $\alpha$ is a learning rate linked to the unconditional stimulus (US) and $\alpha_A$ is the learning rate linked to the CS A. $V_{AB}$ is the compound associative strength for both CS A and B, where B is another CS that sometimes occurs with A.

   A simplified version of the R-W model was used here. The maximum amount of learning $\lambda_U$ that can occur during the trial is simplified to whether the unconditional stimulus, water, shows up or not (1 or 0) since the number of conditional stimuli can only be 1 (only one choice made every trial). The compound stimulus value was not used for the same reason.

   The softmax function used to calculate the probability of choosing rightward is:

   $$p(c(t) = r) = \frac{1}{1 + e^{-\beta(V_r(t) - V_l(t) + b + \kappa a(t-1))}}$$

   $V_l(t)$ is the value associated with the leftward choice by the model, while $V_r(t)$ is for the right. $\beta$ is the inverse-temperature parameter which describes the subjects' confidence (stochasticity) in their decision, $b$ is the bias term, and $\kappa$ implements

the autocorrelation of the previous choice ($a(t-1) = -1$ for a leftward choice and $a(t-1) = 1$ for a rightward choice, different from the quantification in the other part of this study).

This is the softmax function that gives the best performance during another variant of dynamic foraging task for the greatest amount of mice (compared with models without one or more of the $\beta, b, \kappa$ parameters, Bayesian Information Criteria, Bari et al. 2019), thus it is used rather the R-W's softmax function that does not have $b$ and $\kappa$.

- **R-W Simplified**

  Starting from a simplified version of the model where only the learning rate associated with the unconditional stimulus $\alpha$ is utilised. The formula actually looks more similar to the Bush and Mosteller model (Bush and Mosteller 1951a, Bush and Mosteller 1951b), but since there are more than one conditional stimuli in this experiment, it is still categorized as a variant of the more generalized R-W model.

  When the animal chooses the left side, the associated probability is updated as follows:

  $$V_l(t+1) = V_l(t) + \alpha \cdot (R_i - V_l(t))$$

  Similarly for the right side:

  $$V_r(t+1) = V_r(t) + \alpha \cdot (R_i - V_r(t))$$

- **R-W Full**

  The learning rates for the conditional stimulus are added to form the full form of R-W model, and when the animal chooses the left side, the associated probability is updated with:

  $$V_l(t+1) = V_l(t) + \alpha_l \cdot (R_i - V_l(t))$$

  Where:

  $$\alpha_l = \alpha_0 \cdot \alpha$$

  $\alpha_0$ is the learning rate for the conditional stimulus of a leftward wheel turn. $\alpha_0 \cdot \alpha$ is implicit in the modelling process for simplicity during optimization.

  Similarly for the right side:

  $$V_r(t+1) = V_r(t) + \alpha_r \cdot (R_i - V_r(t))$$

- **Decayed RW**

  A "forgetting parameter" is added to the above two models to form the decayed version of **R-W Simplified** and **R-W Full**. Using R-W Full as an example, when the animal chooses the left side, the associated probabilities on both side are updated with:

$$V_l(t+1) = V_l(t) + \alpha_l \cdot (R_i - V_l(t))$$
$$V_r(t+1) = \gamma \cdot V_r(t)$$

Similarly for the right side:

$$V_r(t+1) = V_r(t) + \alpha_r \cdot (R_i - V_r(t))$$
$$V_l(t+1) = \gamma \cdot V_l(t)$$

- **R-W with Motivation**

  This is a group of 4 models that adds a motivation parameter to the above 4 models. Following the observation that animals would be paying less attention to the changes in reward probability if they are not motivated to do the task (very few switches are usually made on Mondays after they have been given free water during the weekend). No additional parameters were added for this version, the learning rate was simply multiplied by the normalized motivation parameter (in the range of [0-1]). Using decayed full R-W as an example, when the animal chooses the left side, the associated probabilities on both side are updated with:

$$V_l(t+1) = V_l(t) + m(t) \cdot \alpha_l \cdot (R_i - V_l(t))$$
$$V_r(t+1) = \gamma \cdot V_r(t)$$

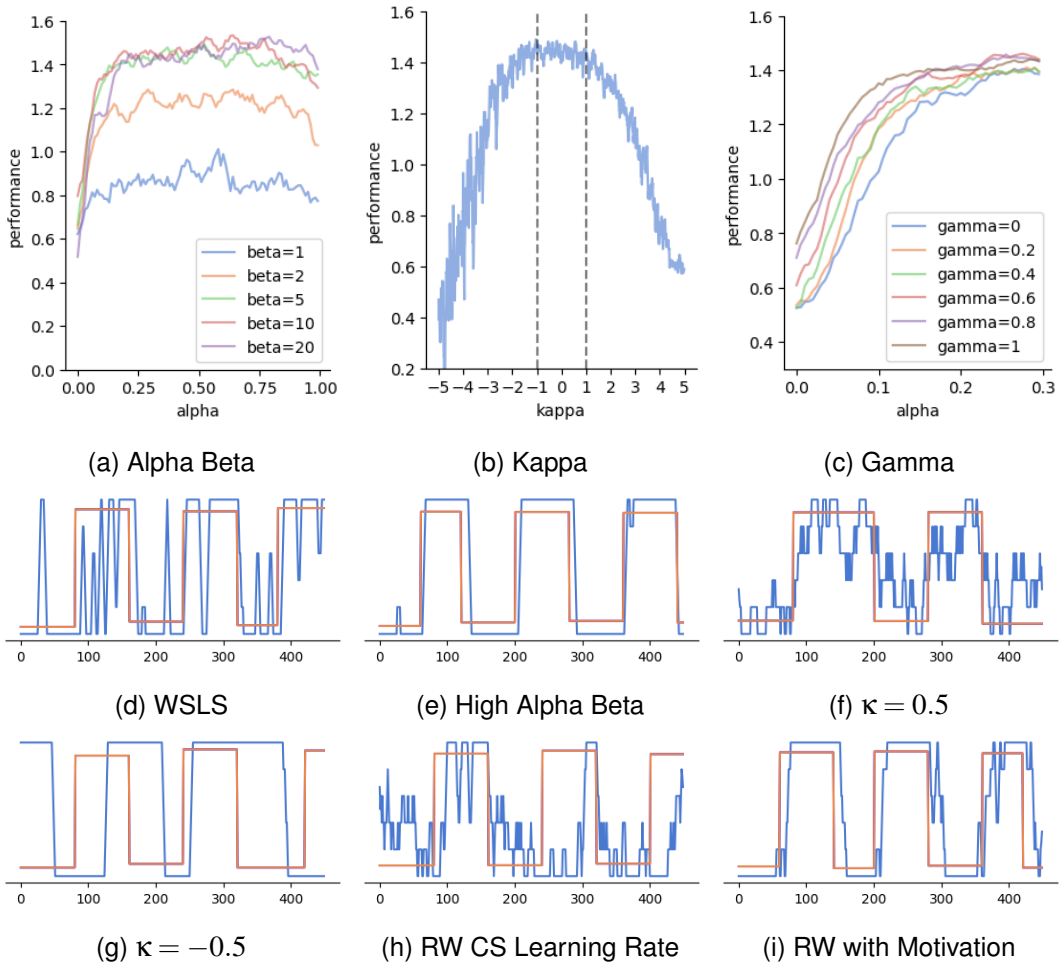
  Where $m(t)$ is the motivation parameter on timestep t.

### 3.2.1 Simulation

The first stage of the simulation optimization pipeline is to simulate the experiment and use the models to make decisions for a better sense of what the models' parameters control and if the models' choices match the animals' behaviour. The switching rule follows the actual task, with switching tests starting to be conducted every 20 trials after the minimum block length was exceeded. A probability switch takes place if 15 out of the previous 20 trials are directed toward the advantageous side.

Figure 3.8d shows an example session simulated with the WSLS model using the default parameter value. It shows reasonable similarities to a clean session only after the win-stay probability is as high as 0.99.

Moving on to the R-W models. The bias term $b$ is relatively straightforward, as it shifts the result of the calculation of the inverse temperature model towards a very positive or negative value, and thus towards a rightward or leftward choice, simulating the inherent bias that an animal might have. However, the influences of the other terms are not as clear. Starting with the unconditional learning rate $\alpha$ and inverse temperature $\beta$ values that are common in all R-W models, their relationship with session performance is shown in Figure 3.8a. Increasing beta and alpha can both nudge the animals towards making a more rational decision and increase the simulated performance, but only to




(a) Alpha Beta (b) Kappa (c) Gamma

(d) WSLS (e) High Alpha Beta (f) $\kappa = 0.5$

(g) $\kappa = -0.5$ (h) RW CS Learning Rate (i) RW with Motivation

Figure 3.8: Parameters in Simulation

a certain extent where the simulated agent almost instantly makes a task switch when noticing that the outcome hasn't been rewarding, shown in Figure 3.8e, simulated with $\alpha = 0.4, \beta = 5$ using RW-Simple model.

With $\alpha = 0.4, \beta = 5$ and RW-Simple model, $\kappa$'s effect on task performance was also investigated. Figure 3.8b shows that $\kappa$ has very small effects on task performance in the range [-0.2, 0.3]. However, an agent with $\kappa < -0.2$ shows high interest in exploring other options and choosing the side opposite of the choice in the last trial (Figure 3.8f), while an agent with $\kappa > 0.3$ shows a strong preference towards sticking with its previous choice (Figure 3.8g). $\gamma$ on the other hand does not appear to bring any additional traits into the behaviour of the simulating agents. Figure 3.8c shows the performances of using different decay coefficient $\gamma$ with $\beta = 5$ in a RW-Simple-Decay model. Neither overall performance nor choices made by the decayed agent show anything beyond the effect of a decreased learning rate, casting some doubt on the necessity of this term.

The effects of the learning rates associated with conditional stimuli rightward wheel turn and leftward wheel turn $\alpha_0, \alpha_1$ are less intuitive. Figure 3.8h shows an example with $\alpha_l = 0.01, \alpha_r = 0.4, \beta = 5$ with RW-Full model. As you can see, having a lower learning rate for the left side actually makes the model more reluctant to switch to the

right side, as updating is slow when making leftward choices. And the model quickly switches back to the left side after it has a couple of unrewarded rightward choices. This is great for simulating animals' "sticky" behaviour where they show a strong preference towards a certain side of wheel turn.

Motivation models were simulated using average motivations calculated at the beginning of the chapter. Figure 3.8i shows messier choices towards the end of the session with lower motivation, which is also observable in the animals' behaviour.

### 3.2.2 Parameter Recovery

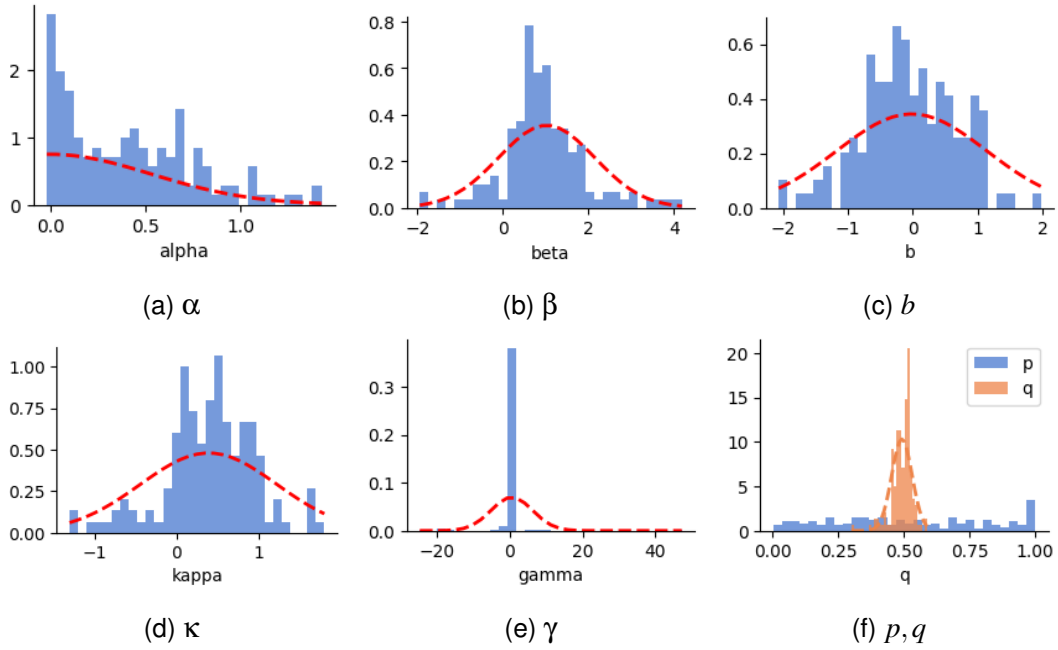
The models are fitted on the behaviour data by identifying the set of parameters that result in the lowest negative log-likelihood. scipy.optimize.minimize was used in this study for optimization. To reduce the risk of encountering local minima, the optimizer commenced from ten distinct sets of random guesses during fitting. Note that during model fitting, NaN trials from behaviour data were removed, since no candidate models can simulate such behaviour.

To evaluate the efficacy of the model fitting process in revealing the parameters that underlie the behavior data, the parameter recovery process was utilized. This involved generating a set of behaviour data with known parameters and determining whether the fitted parameters could recover the simulating parameters.

The initial step in the simulation process is to determine the ranges and distributions for all possible parameters. Some parameters of interest, including win-stay as well as lose-probability probability $p, q$, learning rate $\alpha$, inverse temperature $\beta$, and decay parameter $\gamma$, have predefined bounds, specifically: $\alpha, \gamma, p, q \in [0, 1]; \beta \geq 0$ (Otto et al. 2010, Bari et al. 2019). However, the ranges for other parameters are undefined, and the distribution of all parameters in animal behaviour for this task is unknown. To have a more precise distribution of parameters during simulation, the RW-Simple-Decayed model was fitted onto training data of PLED05-08, and the resulting parameters are displayed in Figure 3.9. Outliers were eliminated based on the MATLAB definition of outliers in the isoutlier function, in which data points that deviate more than 3 standard deviations from the mean are deemed outliers. The results obtained without the outliers were satisfactory, with the exception of $\gamma$ in Figure 3.9e, which had a large standard deviation. All parameters exhibited some degree of normal distribution, with $\alpha$ appearing to be half of a normal distribution. Thus, a normal distribution was fitted to each parameter as a red line, which will be used to generate simulating parameters, while a truncated version based on the previously defined bounds will be employed for $\alpha$, $\beta$, and $\gamma$.

It should be noted that in order to minimize the influence of other parameters, each parameter's result is obtained using simulation or fitting with the smallest model that includes the parameter. For instance, the parameters $\alpha$, $\beta$, $\kappa$, and $b$ were simulated using the RW-Simple model, whereas $\gamma$ was simulated using the RW-Simple-Decayed model.

The distributions of $p, q$ in the WSLS model are depicted in Figure 3.9f. The distribution of $p$ seems to be uniform, whereas $q$ exhibits a normal distribution (with the fitted distribution displayed as a line in the matching colour).




(a) α (b) β (c) $b$

(d) κ (e) γ (f) $p, q$

Figure 3.9: Parameters Distribution in Training Data

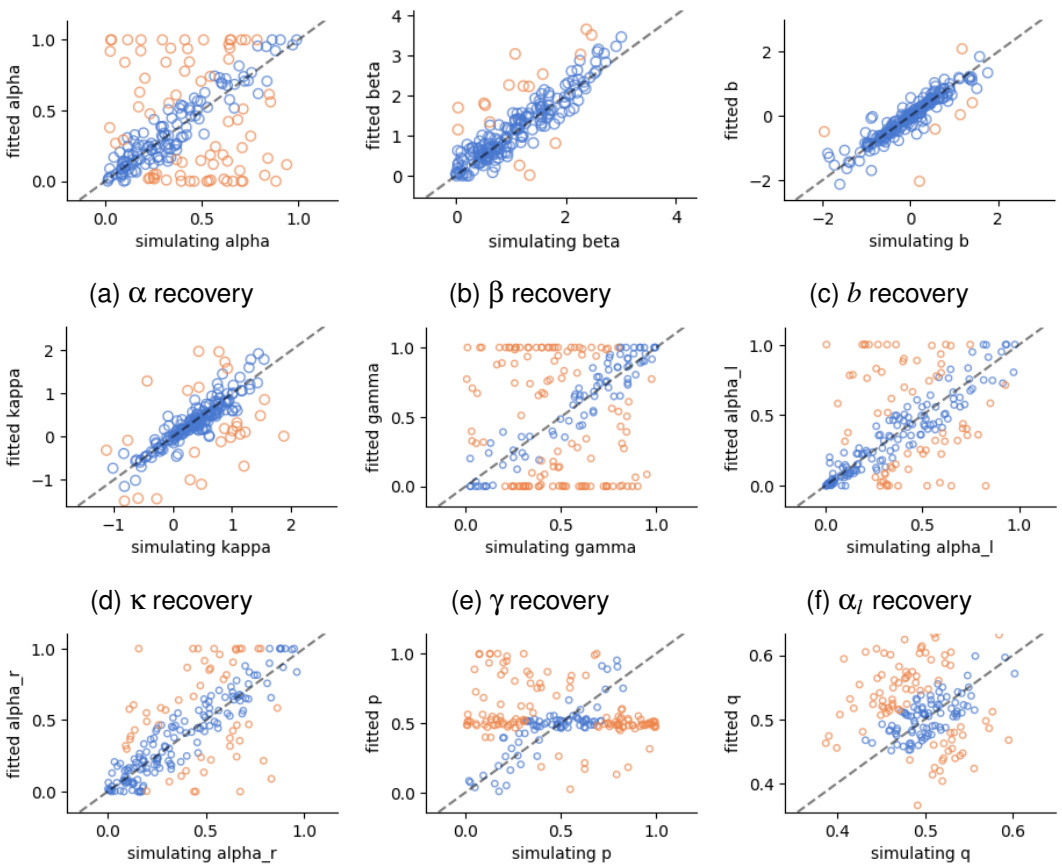
The recovery results of the parameters β, $b$, and κ were satisfactory within their respective ranges, with κ exhibiting superior performance particularly when it is smaller than 0.5. The recovery of the learning rate parameter α (Figure 3.10a) demonstrated acceptable results, as did $\alpha_l$ (Figure 3.10f) and $\alpha_r$ (Figure 3.10g). Notably, the recovery for learning rates within the range [0, 0.2] was significantly better. This is consistent with the simulation results, which demonstrated that performance often plateaus when the learning rate exceeds 0.2, potentially contributing to the difficulty in parameter recovery beyond this threshold.

In contrast, Figure 3.10e indicates that recovery of the γ parameter was poor within its range. As a result, the decayed models were excluded from subsequent stages of analysis, as their inclusion would not yield useful information for fitting real-world behavioural data.

The outcomes of the recovery process for both $p$ (as shown in Figure 3.10h) and $q$ (as demonstrated in Figure 3.10i) were unsatisfactory. The fitted $p$ exhibited a clustering around 0.5, while $q$ showed a slightly cleaner recovery. Despite this, the model was retained for a wider range of models. It's important to note that the parameters obtained through the stochastic WSLS models should now be used in other analysis with caution, as there is a high likelihood that they may not accurately represent the underlying process.

### 3.2.3 Model Recovery

In model comparison, our goal is to determine which model, out of a set of possible models, is most likely to have generated the data. And similar to parameter recovery, model recovery is a controlled simulated process that assesses if the NLL (negative log




(a) α recovery (b) β recovery (c) $b$ recovery

(d) κ recovery (e) γ recovery (f) $\alpha_l$ recovery

(g) $\alpha_r$ recovery (h) $p$ recovery (i) $q$ recovery

Figure 3.10: Parameters Recovery Result

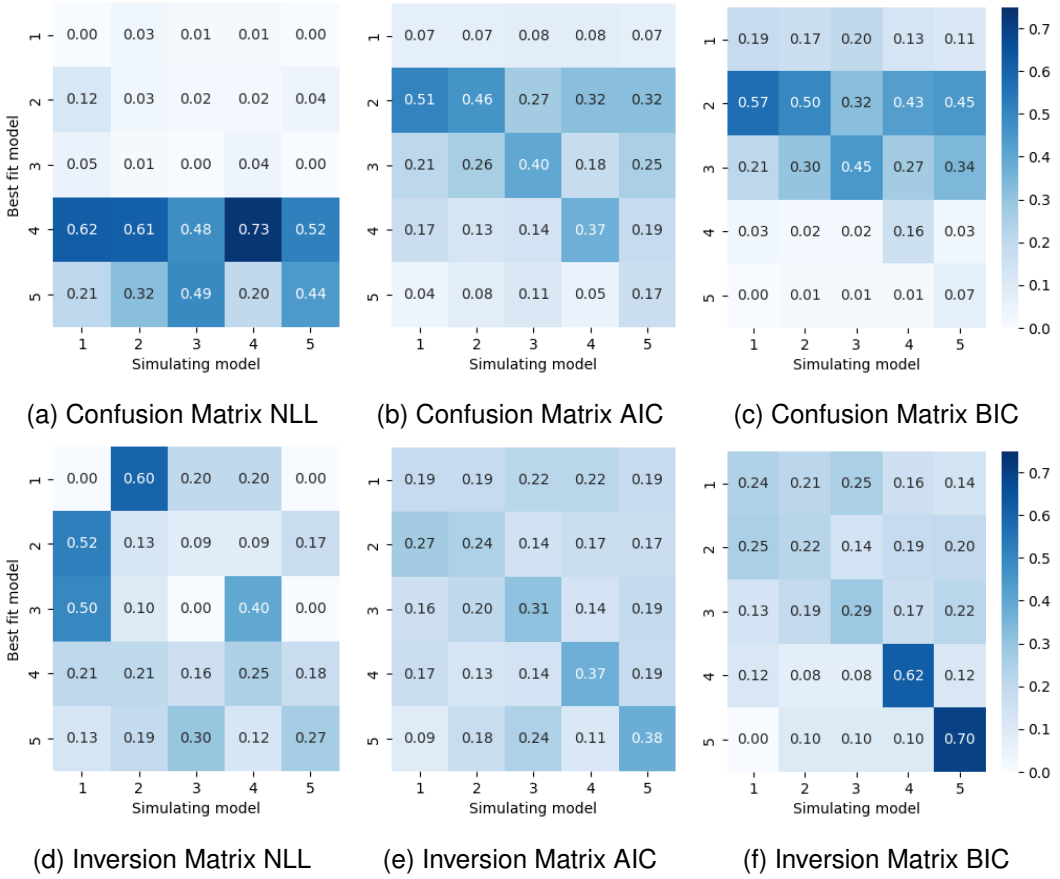
likelihood) can accurately describe how well a model fits the data so that the model generating the data can be uncovered.

Five models were utilized in this stage after removing the decayed versions. Each of these models generated 100 sets of behavior data using the same range acquired from the training data and attempted to fit onto all 500 sets of simulated data.

To enhance visual clarity during visualization, numerical indices were assigned to the models in the illustrations. The models were ranked in increasing complexity and assigned 1-based indices. The models WSLS, RW-Simple, RW-Simple-Motivation, RW-Full, and RW-Full-Motivation were assigned indices 1 through 5, respectively.

Three different metrics, namely, negative log likelihood (NLL), Akaike Information Criteria (AIC), and Bayesian Information Criteria (BIC), were employed while fitting the models.

Figure 3.11a, 3.11b and 3.11c shows the resulting confusion matrix, where each cell contains the conditional probability $P(BestFit|Simulating)$. In plain text, it means the probability of a model fits onto data simulated by itself the best. Figure 3.11a shows that using only NLL made it challenging to recover simpler models by reporting the best fitting data for most sets of data as the more complex RW-Full and RW-Full Motivation.




(a) Confusion Matrix NLL (b) Confusion Matrix AIC (c) Confusion Matrix BIC

(d) Inversion Matrix NLL (e) Inversion Matrix AIC (f) Inversion Matrix BIC

Figure 3.11: Model Recovery Result

In contrast, BIC in Figure 3.11c was too stringent on more complex models. AIC in Figure 3.11b provided a decent recovery for all RW models and was, therefore, the most balanced option.

The subfigures 3.11d, 3.11e, and 3.11f, on the other hand, depict the probability of direct interest when fitting the models on data. Each cell contains the conditional probability $P(Simulating|BestFit)$, which shows how confident one can be that the best-fitted model is the one that simulated the data. AIC in Figure 3.11e provides the most balanced performance for all models, but BIC (depicted in Figure 3.11f) is not far behind AIC on simpler models and has much stronger confidence for more complex models.

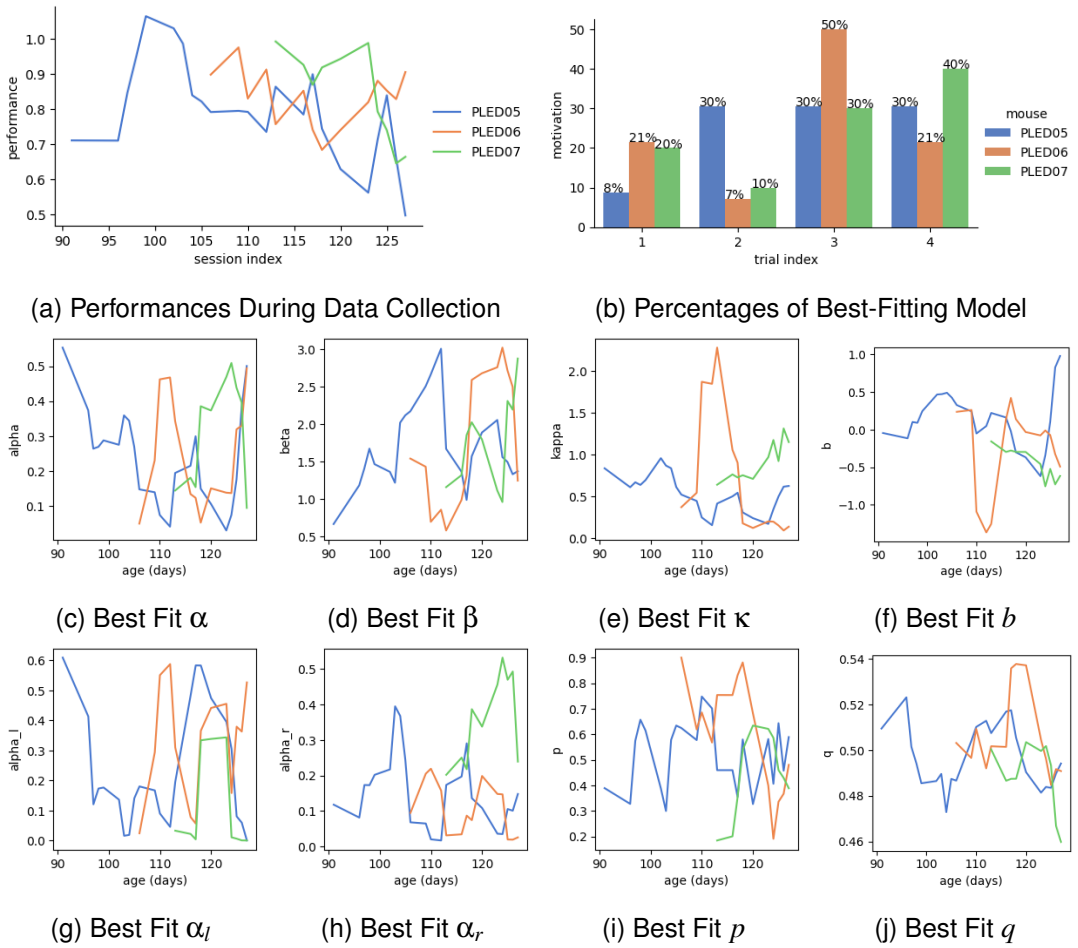
According to the analysis, WSLS models were not well recovered by any of the three metrics. The data simulated with WSLS is most frequently fitted as an RW-Simple Model, suggesting that the version of WSLS model used in this study may not have the ability to capture the distinct difference between associative and rule-based decision making.

In conclusion, I found AIC to be the most suitable criteria for a broader range of models and decided to use it when comparing the models on real animals' behavior.

## 3.3 Testing Main Hypothesis

In this section, the main hypothesis of the study was tested using data collected from three participating animals that passed the final benchmakr, with PLED05 and PLED07 being the F7/F7 disease-type animals, while PLED06 was healthy control. It is important to note that all illustrations presented in this section were produced while the researcher was still blinded to the animals' genotypes to prevent introducing bias.

While the sample size is currently too small for significance testing, some preliminary results can still provide insight into possible directions for future research.




(a) Performances During Data Collection (b) Percentages of Best-Fitting Model

(c) Best Fit α (d) Best Fit β (e) Best Fit κ (f) Best Fit $b$

(g) Best Fit $\alpha_l$ (h) Best Fit $\alpha_r$ (i) Best Fit $p$ (j) Best Fit $q$

Figure 3.12: Main Hypothesis Tests, Legend in Figure 3.12a Applies to All Lineplots Figure 3.12c to 3.12j

### 3.3.1 Vascular Dementia's Influence on Task Training and Performance

Figure 3.12a shows the performances during the data collection stage of the three animals, plotted against the age of the animals at the session date and smoothened over three sessions for better visual clarity. An obvious drop in performance can be noticed in the two F7/F7 animals PLED05 and 07, while the performance of PLED06 remained

stable. However, it is worth noting that the number of sessions and animals is too few to perform any informative tests, and no conclusive results can be shown until the animals' brains are dissected and tested with markers. Nevertheless, these preliminary results are encouraging.

On the other hand, the fact that the healthy control animal PLED08 was unable to finish the training indicates that the equipment and task parameters may need further refinement to accommodate a wider range of individual differences. However, this does not invalidate the overall viability of the setup, since Figure 3.5 shows that PLED08's performance was increasing throughout training.

### 3.3.2 Vascular Dementia's Influence on Best-Fitting Model and Model Parameters

The percentage of best-fitting models for each animal during the data collection phase using AIC is demonstrated in Figure 3.12b. The same complexity-based indices used in subsection 3.2.3 were employed for cleaner visualization. Although no noticeable difference was observed between PLED06 and the two F7/F7 animals, PLED06 had more sessions best fitted by the RW-Simple-Motivation model. The plot also shows that the motivation parameters provided much less information in the RW-Full model than the RW-Simple model, with a higher percentage of sessions being best fitted using RW-Simple-Motivation, and none were best fitted to the RW-Full-Motivation model.

The best-fitting parameters were acquired using the same simplest model principle mentioned in subsection 3.2.2, and are plotted against the animals' age at the date of the session and also smoothened over three sessions, as shown in Figure 3.12c to 3.12j. However, no significant differences can be seen between the normal control group and F7/F7 animals.

# Chapter 4

# Conclusion

In conclusion, although the modeling did not yield satisfactory results, this study was still productive. Three out of the four officially participating animals were able to complete the training within the designated time frame before disease onset, demonstrating that the experiment design and custom setup are viable for larger-scale deployment, albeit some further refinement may be necessary before it can be published as a standard.

Encouraging results in support of the hypothesis that trained disease-type animals exhibit declined performances were demonstrated in subsection 3.3.1. However, limited data prevented definitive conclusions from being drawn. To produce a complete and publishable work, at least an additional year and a half of effort will be necessary, assuming that the setup can be replicated, and concurrent training can take place on both setups.

Firstly, modifications to the experiment design may be required, particularly with respect to the difficulty of training_2 and the data collection version of the task. Currently, animals did not appear to require any additional learning to handle the more challenging version. Additionally, equipment improvements may be necessary to enhance reliability and standardization, making the setup easier to replicate.

Secondly, a significantly larger sample size is necessary. Power analysis was conducted, indicating that if there is a 30% difference in pericyte loss between the two groups, 16-18 animals will be required in each group to yield informative and significant results.

If current models still cannot produce informative result with more substantial data, more sophisticated models may be required. Including TD learning or a model-based learning model in the model-fitting process can lead to interesting discussions. To date, nearly all modeling work on reversal learning tasks has been focused on R-W-type reinforcement learning models that optimize immediate rewards rather than session rewards. Additionally, a much stronger rule-based decision-making model is necessary, as the stochastic WSLS model used in this study did not perform well during the simulation stages.

Crainiotomy and electrophysiology recording in specific area responsible for decision making, for example mPFC (medial prefrontal cortex) and striatum (Euston, Gruber,

and McNaughton 2012, Goulet-Kennedy, Labbe, and Fecteau 2022), may be needed for the modelling result to be more convincing. And histological analysis with pericyte markers is certainly necessary to ensure that the disease type animal does have pericyte detachment.

Finally, an attempt was made to automate the simulation optimization pipeline, and all models' classes strictly followed one template of required functions for this purpose. However, human intervention was still required when determining the range of the simulating parameters. If this challenge can be overcome, model comparison can be conducted at a much faster pace.

# Bibliography


Abbott, Laurence F. and Peter Dayan (2001). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. MIT Press.

Bari, Bilal A. et al. (2019). "Stable Representations of Decision Variables for Flexible Behavior". In: *Neuron* 103.5, 922–933.e7. ISSN: 0896-6273. DOI: https://doi.org/10.1016/j.neuron.2019.06.001. URL: https://www.sciencedirect.com/science/article/pii/S089662731930529X.

Barkus, Chris et al. (2022). "Refinements to rodent head fixation and fluid/food control for neuroscience". In: *Journal of Neuroscience Methods* 381, p. 109705. ISSN: 0165-0270. DOI: https://doi.org/10.1016/j.jneumeth.2022.109705. URL: https://www.sciencedirect.com/science/article/pii/S016502702200231X.

Birrell, James M and Victoria J Brown (2018). "Assessment of intradimensional/extradimensional attentional set-shifting in rats". In: *Current protocols in neuroscience* 82, e51.

Bush, Robert R. and Frederick Mosteller (1951a). "A Mathematical Model for Simple Learning". In: *Psychological Review* 58.5, pp. 313–323. DOI: 10.1037/h0054388.

– (1951b). "A Model for Stimulus Generalization and Discrimination". In: *Psychological Review* 58.6, pp. 413–423. DOI: 10.1037/h0054576.

Codd, E.F. (1971). "Further Normalization of the Data Base Relational Model". In: *IBM Research Report RJ909*.

Denrell, Jerker and Chengwei Liu (2012). "Working-memory load and temporal myopia in dynamic decision making". In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 38.1, p. 164.

Ding, Ren et al. (2020). "Loss of capillary pericytes and the blood–brain barrier in white matter in poststroke and vascular dementias and Alzheimer's disease". In: *Brain Pathology* 30.6, pp. 1087–1101.

Euston, David R., Aaron J. Gruber, and Bruce L. McNaughton (2012). "The Role of Medial Prefrontal Cortex in Memory and Decision Making". In: *Neuron* 76.6, pp. 1057–1070. ISSN: 0896-6273. DOI: https://doi.org/10.1016/j.neuron.2012.12.002. URL: https://www.sciencedirect.com/science/article/pii/S0896627312011087.

Gazzaniga, Michael S, Richard B Ivry, and George R Mangun (2019). *Cognitive Neuroscience: The Biology of the Mind*. 5th. W. W. Norton & Company, pp. 534–536.

Glimcher, Paul W. (2011). "Understanding dopamine and reinforcement learning: The dopamine reward prediction error hypothesis". In: *Proceedings of the National Academy of Sciences* 108.supplement_3, pp. 15647–15654. DOI: 10.1073/pnas.1014269108. eprint: https://www.pnas.org/doi/pdf/10.1073/

pnas.1014269108. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1014269108.

Gluck, Mark A and Gordon H Bower (2014). "A comparison model of reinforcement-learning and win-stay-lose-shift decision-making processes: A tribute to WK Estes". In: *Journal of Mathematical Psychology* 60, pp. 21–32.

Goulet-Kennedy, Julie, Sara Labbe, and Shirley Fecteau (2022). "The involvement of the striatum in decision making". In: *Dialogues in Clinical Neuroscience*.

Hammoudeh, Ahmad (Feb. 2018). "A Concise Introduction to Reinforcement Learning". In: DOI: 10.13140/RG.2.2.31027.53285.

Herrnstein, R. J. (1961). "RELATIVE AND ABSOLUTE STRENGTH OF RESPONSE AS A FUNCTION OF FREQUENCY OF REINFORCEMENT1,2". In: *Journal of the Experimental Analysis of Behavior* 4.3, pp. 267–272. DOI: https://doi.org/10.1901/jeab.1961.4-267. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1901/jeab.1961.4-267. URL: https://onlinelibrary.wiley.com/doi/abs/10.1901/jeab.1961.4-267.

Jara-Rizzo, María F et al. (2020). "Decision-making inflexibility in a reversal learning task is associated with severity of problem gambling symptoms but not with a diagnosis of substance use disorder". In: *BMC Psychology* 8.1, p. 120. DOI: 10.1186/s40359-020-00482-6.

Kahneman, Daniel and Amos Tversky (2013). "Choices, Values, and Frames". In: *HANDBOOK OF THE FUNDAMENTALS OF FINANCIAL DECISION MAKING Part I*. World Scientific Publishing Co. Pte. Ltd. Chap. 16, pp. 269–278. URL: https://EconPapers.repec.org/RePEc:wsi:wschap:9789814417358_0016.

Kalaria, Raj N. (2018). "The pathology and pathophysiology of vascular dementia". In: *Neuropharmacology* 134. Cerebral Ischemia, pp. 226–239. ISSN: 0028-3908. DOI: https://doi.org/10.1016/j.neuropharm.2017.12.030. URL: https://www.sciencedirect.com/science/article/pii/S0028390817306275.

Kumar, Ashutosh et al. (2021). "Single-Cell Analysis of Blood-Brain Barrier Response to Pericyte Loss". In: *Circulation Research* 128.8, pp. 1076–1086. DOI: 10.1161/CIRCRESAHA.120.317473. eprint: https://www.ahajournals.org/doi/pdf/10.1161/CIRCRESAHA.120.317473. URL: https://www.ahajournals.org/doi/abs/10.1161/CIRCRESAHA.120.317473.

Ludvig, Elliot, Marc Bellemare, and Keir Pearson (Jan. 2011). "A Primer on Reinforcement Learning in the Brain: Psychological, Computational, and Neural Perspectives". In: pp. 111–144. ISBN: 9781609600235. DOI: 10.4018/978-1-60960-021-1.ch006.

Martin, Howard (2020). "23.3: Back EMF in an electric motor". In: *Physics LibreTexts*. URL: https://phys.libretexts.org/Bookshelves/University_Physics/Book%3A_Introductory_Physics_-_Building_Models_to_Describe_Our_World_(Martin_Neary_Rinaldo_and_Woodman)/23%3A_Electromagnetic_Induction/23.03%3A_Back_EMF_in_an_electric_motor.

Mizunami, Makoto, Kanta Terao, and Beatriz Alvarez (2018). "Application of a Prediction Error Theory to Pavlovian Conditioning in an Insect". In: *Frontiers in Psychology* 9. ISSN: 1664-1078. DOI: 10.3389/fpsyg.2018.01272. URL: https://www.frontiersin.org/articles/10.3389/fpsyg.2018.01272.

Montagne, Axel et al. (Mar. 2018). "Pericyte degeneration causes white matter dysfunction in the mouse central nervous system". In: *Nature medicine* 24.3, pp. 326–337. ISSN: 1078-8956. DOI: 10.1038/nm.4482. URL: https://europepmc.org/articles/PMC5840035.

Otto, A Ross et al. (2010). "Regulatory fit and systematic exploration in a dynamic decision-making environment." In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 36.3, p. 797.

Paré, William P. and Gary B. Glavin (1986). "Restraint stress in biomedical research: A review". In: *Neuroscience and Biobehavioral Reviews* 10.3, pp. 339–370. ISSN: 0149-7634. DOI: https://doi.org/10.1016/0149-7634(86)90017-5. URL: https://www.sciencedirect.com/science/article/pii/0149763486900175.

Perandrés-Gómez, Ana et al. (2021). "Decision-making (in)flexibility in gambling disorder". In: *Addictive Behaviors* 112, p. 106534. ISSN: 0306-4603. DOI: https://doi.org/10.1016/j.addbeh.2020.106534. URL: https://www.sciencedirect.com/science/article/pii/S030646032030664X.

Peterson, D.A. et al. (2009). "Probabilistic reversal learning is impaired in Parkinson's disease". In: *Neuroscience* 163.4, pp. 1092–1101. ISSN: 0306-4522. DOI: https://doi.org/10.1016/j.neuroscience.2009.07.033. URL: https://www.sciencedirect.com/science/article/pii/S0306452209012068.

Procter, Tessa V, Anna Williams, and Axel Montagne (2021). "Interplay between brain pericytes and endothelial cells in dementia". In: *The American Journal of Pathology* 191.11, pp. 1917–1931.

Series, Peggy, ed. (Nov. 2020). *Computational Psychiatry: A Primer*. English. MIT Press. ISBN: 9780262044592.

Shi, Haoshen et al. (2020). "Identification of early pericyte loss and vascular amyloidosis in Alzheimer's disease retina". In: *Acta neuropathologica* 139, pp. 813–836.

Stuart, Sarah A. et al. (2013). "A translational rodent assay of affective biases in depression and antidepressant therapy". In: *Neuropsychopharmacology* 38.9. Cited by: 63; All Open Access, Green Open Access, Hybrid Gold Open Access, pp. 1625–1635. DOI: 10.1038/npp.2013.69. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84880332936&doi=10.1038%2fnpp.2013.69&partnerID=40&md5=51350850bc2d3460fc33f0697e8b2210.

Sutton, Richard S and Andrew G Barto (2018). *Reinforcement learning: An introduction*. MIT press.

Sweeney, Melanie D., Saravanan Ayyadurai, and Berislav V. Zlokovic (2018). "Pericytes of the neurovascular unit: key functions and signaling pathways". In: *Journal of Neurochemistry* 146.6, pp. 660–672. DOI: 10.1111/jnc.15217. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/jnc.15217. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/jnc.15217.

Thorndike, E. L. (1898). "Animal Intelligence: An Experimental Study of the Associative Processes in Animals". In: *Psychological Review* 5.5, pp. 551–553. DOI: 10.1037/h0067373.

Uemura, Maiko T et al. (2020). "Brain microvascular pericytes in vascular cognitive impairment and dementia". In: *Frontiers in aging neuroscience* 12, p. 80.

Wilson, Robert C and Anne GE Collins (Nov. 2019). "Ten simple rules for the computational modeling of behavioral data". In: *eLife* 8. Ed. by Timothy E Behrens, e49547.

ISSN: 2050-084X. DOI: 10.7554/eLife.49547. URL: https://doi.org/10.7554/eLife.49547.

Zhang, C Eleana et al. (2019). "Blood–brain barrier leakage in relation to white matter hyperintensity volume and cognition in small vessel disease and normal aging". In: *Brain imaging and behavior* 13, pp. 389–395.

# Appendix A

# Supplementary figures

## A.1 Performance Comparison

This contains Figure 3.2 for all animals.

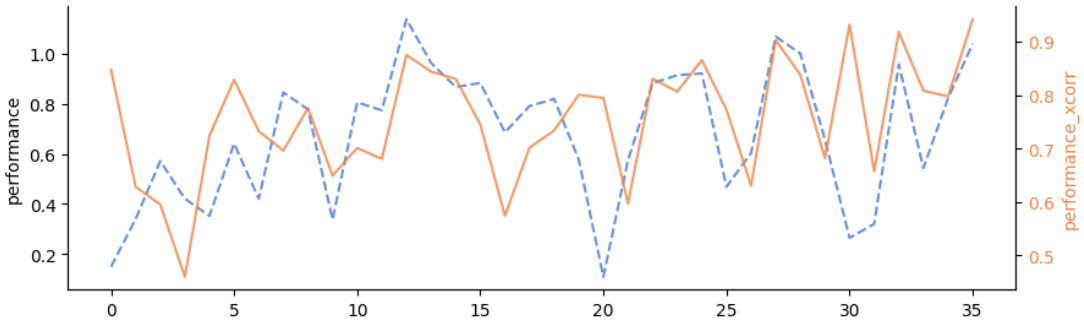



Figure A.1: PLED05

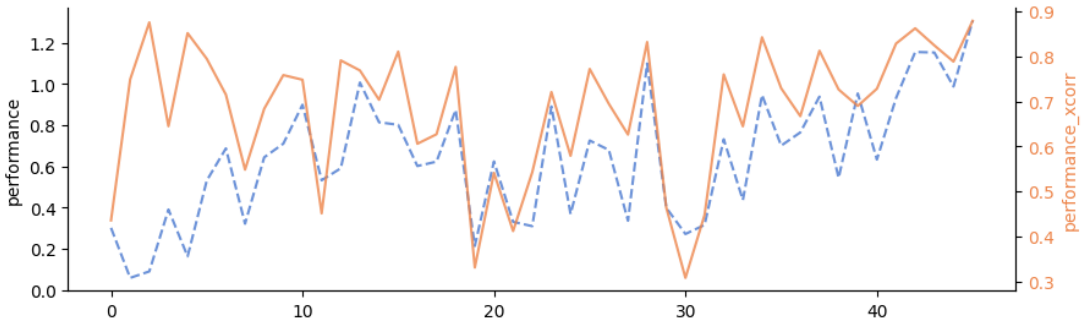



Figure A.2: PLED06

Similar conclusion could be drawn from all animals other than PLED08.

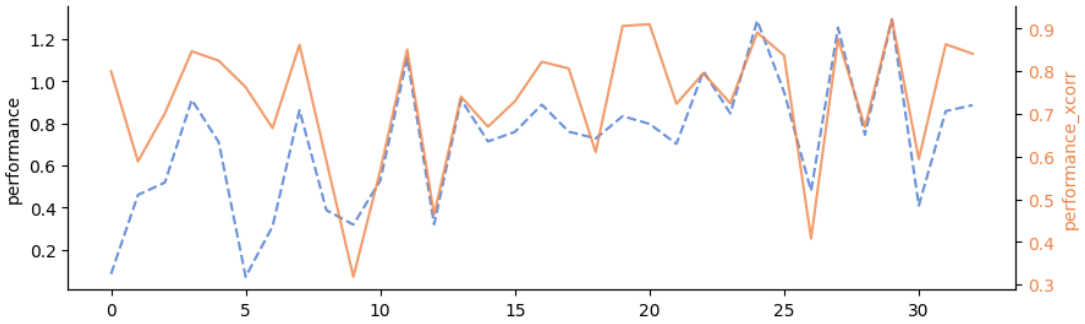



Figure A.3: PLED07

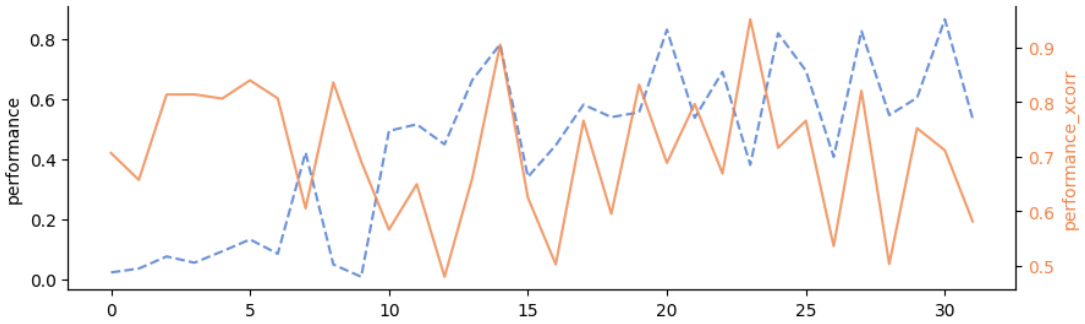



Figure A.4: PLED08

# Appendix B

# Guide to Repositories

## B.1 Dynamic Foraging Repository

The bullet points below give a quick overview of the packages in this project, which are located under the lib directory:

- **Benchmark**

  *'benchmark.py'* is used for benchmarking the animal's per trial performance during the experiment, and deciding whether they can move on to the next stage or not, with "evaluate.py" containing all the helper functions for quantitatively analyse an animal's performance in a session.

- **Hardware**

  All of the functions used to interact with the hardware components can be found here. For now it only contains the valve control functions that delivers the reward to the animal. For future reference, code for communicating with electrophysiology recording should also be located here.

- **Database**

  All experimental data are automatically uploaded to the Microsoft Azure SQL database server. Microsoft Azure Educate offers $100 credits per year, sufficient for the purpose of this experiment. A local copy of csv file for each session is also generated for backup.

  The functions needed to interact with Azure databases, as well as creating the local copy of csv files are included in the Database package. All the queries used to interact with the database are contained in functions of *'queries.py'*, while *'modify.ipynb'* helps make database interaction easier using said functions.

  Figure B.1 introduces the structure of the database, and each relation is designed to meet the requirement of the third normal form(3NF) (Codd 1971).

  The **"mice"** table contains all animals' basic information - its name (mouse_code), date of birth, their experimental stages (will be mentioned in section 2.1), as well

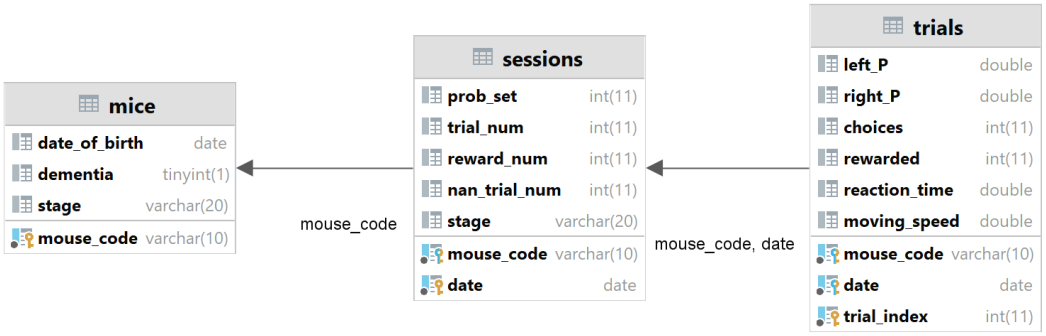



Figure B.1: database structure

as whether they have dementia or not. The *'dementia'* field remains None until the end of data collection. This information is only known to Dr. Gan to keep the training process with as little bias as possible.

The **"sessions"** table includes the summary information of each session that the animals have done. Most of the fields are self-explanatory, the "prob_set" is the random seed used in the session. The animals goes through all the seeds from 0 to 89. The 3-stage training sessions are also given probability set index for uniformity: *"motor_training": -3; "training_1": -2; "training_2": -1*.

The substantially larger **"trials"** stores information about each trial of all animals' sessions. The fields are introduced in more detail at the beginning of chapter 3.

- ui

  This modules contains the graphical interface for making accessing the system easier for operators unfamiliar with command line. It also contains the code for generating the green cursor for visualizing animal's movement, which in principle is a GUI for the animal.

- visualization

  This is where the live visualization of a session is implemented so that the operator can track the animals' performance qualitatively in real time. The illustation is in the same format as shown in Figure 1.3.

## B.2 Model Comparison Repository

This is a much simpler repository and does not have a complex structure of sub packages.

The "data" folder contains the all the animal's training data, downloaded from the remote database and parsed into csv files that are more easily accessible. The output from the simulations and model fittings are also saved here as .npy files.

The "visualization" script contains the functions that plotted all of the visualizations in chapter 3, while the resulting figures are all stored in "figures" directory.

The "models" directory contains the models used in this study, where all models inherit from the "Model" class so that they can have a uniform interface. All models have the following functions:

- update: update the internal values using the choice made in the last trial as well as whether the agent was rewarded or not
- get_choice: make a choice using the internal values.
- nll: calculate the NLL when fitting onto a set of given data using the given parameter values
- fit: find the parameters that provide the best fit to the given data

A "Task" class was also written to control the switching rules and the reward probabilities in the environment.

Three ipython notebooks were written for the analysis in premodelling analysis, simulation optimization pipeline, as well as main hypothesis testing. Their names should be self-explanatory.

# Appendix C

# Additional Information

## C.1 More on Session Lengths

A 45 minutes time limit was also enforced for all the training stages, to further make sure that the animals are not being worn out. The time limit was set using previous experience and was seldom exceeded after the first couple of motor training sessions where the animals had a large amount of NaN trials. The time limit also makes it easier for me and my coworkers to arrange the training sessions around the lectures and other events since they now take a predictable amount of time.

To account for the large individual variances, an offset value is added on top of the set session length so that each animal will have the sessions more tailored to their personalities. The offset is in the range of $\pm 100$, and is incremented or decremented by 20 at a time. The offsets are initialised to 0 and are updated according to the update rules:

- During a session, if an animal manages to finish the session in under 40 minutes **and** has a session nan-percentage lower than 5%, the session offset will be incremented by 20.
- During a session, if an animal cannot finish the session in under 45 minutes **or** has a session nan-percentage higher than 10%, the session offset will be decreased by 20.

The offset is only applied during training_1 and trainig_2 stages, and is reset after a stage transition. A very high nan percentage is expected when the animals first started to learn to use the wheel during motor training and I want to leave an ample amount of time for the mice to familiarise with the device. As a result, no trial-number deduction is applied during the motor training stage.

Also, note that the performance is still benchmarked with the trials exceeding the set session length limit being cut off to ensure fairness across different animals. For example, during training_1, even though an animal can do 500 trials in a session, we only take the first 400 trials into consideration when benchmarking its performance. Since having more trials could mean the possibility of doing more switches, which will

introduce a bias when using my performance metric.