ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

University of Zurich[UZH]

ETH Zürich
Department of Health Sciences and Technology
Master in Health Sciences and Technology
Major in Neurosciences

Reward-directed Spatial Observational Learning in Mice

Master Thesis

In partial fulfilment of the requirements for the Master of Science ETH
in Health Sciences and Technology (MSc ETH HST)

Submitted by:
Yihui Du

Supervisors:
Roman Boehringer
Eduarda Streit Morsch
Prof. Dr. Benjamin F. Grewe
Institute of Neuroinformatics (INI), University of Zürich, ETH Zürich

Tutor:
Prof. Dr. Denis Burdakov
Department of Health Sciences and Technology, ETH Zürich

Zürich, August 2023

Abstract

Observational learning, the ability of animals to acquire direct experience through visual observation of their peers, is a vital survival strategy. While prior studies on observational learning have predominantly concentrated on spatial working memory in rats and observational fear learning in mice, challenges persist. For instance, long-term population recording of single place cell activity with calcium imaging in rats remains problematic, and reward-based learning mechanisms offer more dynamic insights than fear learning. To address these gaps, our study delves into the reward-directed spatial observational learning in mice. We developed a paradigm for spatial observational learning and employed a closed-loop tracking system to train mice to form long-term memory of the reward-directed spatial observational learning tasks on a Cheeseboard maze. The mice are divided into two groups in the experiment: demonstrator (demo) mice and observer (obs) mice. In the demo mouse training session, we initially discovered that the mice's performance in recalling the location of food rewards could improve rapidly, achieving a stable behavioural paradigm after a few days of training. We then implemented behavioural shaping by daily reducing food placement on the maze, aiming to dissociate spatial memory from reward-directed behaviour. Remarkably, we found that even when all trials lacked food rewards on maze, some mice could still accurately shape their spatial memory behaviour after a few days of training. In the observational learning session, we found that some observer mice exhibited improved performance in the testing trials after observing the demo mice. Further, we analyzed the dynamics of the observer mice's head direction at various stages during the observational learning process. We discovered that when the demo mice are reaching the reward zone, the observer mice would significantly face towards the maze, triggered by the cue. Based on the timeseries of head direction, we identified two types of attention during the observational learning process: transferable attention and sustained attention. This project combines the reward-directed spatial learning task with the social observational learning task, substantiating that individual mice can encode and recall long-term spatial memory under the motivation of delayed rewards. In the observational learning experiment, we identified that the head direction can serve as a marker of the observer mouse's attention. These findings provide a viable behavioural paradigm and a quantitative analysis method for future research on observational learning in mice.

Contents

1. Introduction

Observational learning is the capacity to acquire the value of stimuli or optimize behavioural actions by witnessing a similar or relevant experience in another animal (1). Observational learning, and the more general concept of social learning, is an efficient strategy for adapting to changes in environmental conditions (2). The ability to observe, interpret, and learn behaviours and emotions from conspecifics is crucial for survival, as it bypasses direct experience to avoid potential dangers and maximize rewards and benefits (3). Observational learning is a common mode of learning since many animal species including mice are found being able to learn and significantly improve their behavioural performance by observing the behaviour of a conspecific (4). There are many different behavioural paradigms for observational learning across different species, for example, virgin mice can learn to retrieve pups by observing maternal retrieval by experienced mothers (5). We chose a spatial memory paradigm, cheeseboard maze, to study observational learning since it's essential for social animals to know the spatial positions of their conspecifics during reward navigation (6).

In our designed spatial learning and memory task, the hippocampus is thought to be a core brain region because the activities of place cells in the dorsal hippocampus play an important role in animals' awareness of location (7). The formation of new spatial memory relies on the reorganization of assembly firing patterns and plasticity of place cells in the CA1 region of hippocampus (8, 9, 10). The consolidation of spatial memory depends on the reactivation ('replay') of hippocampal place cells that were active during sharp-wave ripples (SWRs) that occur during immobility (11). The 'preplay' of future novel place cell sequences are also found in hippocampal cellular assemblies (12). Besides hippocampus, the anterior cingulate cortex (ACC) is also essential for the integration of social, emotional, and spatial signals during observational learning (13, 14).

Several studies have previously been conducted on the neural mechanisms of observational learning, suggesting that social learning partially shares neural mechanisms with self-experienced learning in both rodents and humans. Many previous studies on the neural mechanisms of observational learning in mice are observational fear learning, which is a form of emotional contagion and is thought to be a basic form of affective empathy that is essential for the well-being of social species (15, 16). The observer mice can develop freezing behaviour by observing the demo mice receiving repetitive foot shocks, which involves affective pain system and ACC-basolateral amygdala (ACC-BLA) circuit (17, 18, 19). The ensembles of CA2 pyramidal neurons are also found active during social exploration of previously unknown conspecifics and reactivating during SWRs.

Instead of observational fear learning, we focus on spatial goal-oriented observational learning, because the ability to recall and navigate to spatial locations associated with rewards is essential to an organism's survival (20, 21). Hippocampus has vasoactive intestinal polypeptide-expressing (VIP) interneurons that support goal-oriented spatial learning (22). Also, a dedicated population of 'reward cells' are found in dorsal and intermediate CA1 that are active at multiple reward sites and exists global remapping in the environment (23, 24). Additionally, neurons modulated by head direction have been reported in hippocampus, where the place cells form goal-oriented vector fields during navigation (25, 26). Some of the existed neural recording experiments during spatial reward-directed observational learning are done in rat and bat brains, revealing that CA1 pyramidal cells in the observer jointly and discretely encode the spatial location of both the self and the other in allocentric coordinates (27, 28). Similarly, the remote awake 'replay' happens during social observation to guide observer rat's spatial decisions (29).

Together, we designed and implemented a spatial reward-directed learning task to investigate observational learning in mice. We conducted behavioural training across three cohorts of mice. To analyze and quantify the learning performance of the individual demo mouse, we calculated the distance of trajectory and the latency between leaving the cage and reaching the reward zone. The results indicated that individual mice could perfectly recall specific spatial locations even in the absence of food reward on maze. Subsequently, for the observational learning task, in addition to using trajectory distance to quantify learning performance, we also employed the DeepLabCut software to track different body parts of the mice (30, 31), and then calculated head directions to characterize the attention dynamics of the observer mice during observational learning (32, 33, 34, 35). The results of this analysis provided additional support for the findings from the spatial observational learning task. Our study presents a novel approach to studying observational learning in mice, combining traditional measures of learning performance with innovative methods for analyzing attention states. This research contributes to our understanding of the behavioural mechanisms underlying observational learning and lays a solid foundation for future studies in this area.

2. Materials and Methods

2.1. Subjects

All procedures were authorized by the veterinary office of the Canton Zurich, Switzerland, and are in agreement with the guidelines published in the European Communities Council Directive of November 24, 1986 (86/609/EEC).

To conduct our spatial reward-oriented observational learning task, we utilized young adult C57BL/6j mice, aged between 4 weeks to 3 months. A total of three cohorts of animal experiments were performed. For each cohort, we slightly adjusted some of the parameters to optimize behavioral performance. In each cohort, we randomly paired group-housed six (cohorts 1 and 2) or four (cohort 3) male (cohorts 1) or female (cohort 2 and 3) mice as demonstrator and observer, and trained them for the entire 21-day learning task batch. The mice in each cohort were housed in a large cage which is designed to comfortably accommodate up to 10 mice.

During the entire 21-day experimental period, to ensure that the mice were motivated to participate in the reward-oriented behavioural task, mice were subjected to food scheduling. During this food scheduling period, mice only had access to *ad libitum* food for a 4-hour time window each day (from 1 PM to 5 PM). During the behavioural training and testing experiments (from 9 AM to 1 PM), mice were given 14mg dustless and odourless sucrose pellets (Bio-Serv) as food rewards. Water was provided *ad libitum* at all times, except when the mice were participating in the task. Throughout the entire experimental period, we monitored the daily weight changes of the mice before and after food restriction. If a mouse's weight decreased by more than 10% of its original weight, it would be excluded from the experiment. Except during the experimental period, the mice were usually housed in the animal facility environment with a regular wake-sleep cycle, where lights were on from 6 am to 6 pm and off from 6 pm to 6 am.

2.2. Cheeseboard maze task

We employed an adapted cheeseboard maze, a 60x60 cm square board functionally similar to the Morris water maze, to test the spatial memory of rodents (8). The Cheeseboard maze is characterized by 36 small holes uniformly distributed at 10 cm apart, in which food pellets can be placed (Figure 1). Mice cannot see the location of the hidden food rewards within their parallel field of vision. Consequently, they must rely on the visual cues from the surrounding environment to memorize and recall the location of food rewards. As the cheeseboard maze does not require immersing the mice in water, it has several advantages over the Morris water maze. These include ease of operation, the capacity to increase the number of experimental trials, and the fact that the animals' behaviour is driven by rewards rather than stress. Furthermore, it is compatible with subsequent calcium imaging of place cells in hippocampus with miniscope.

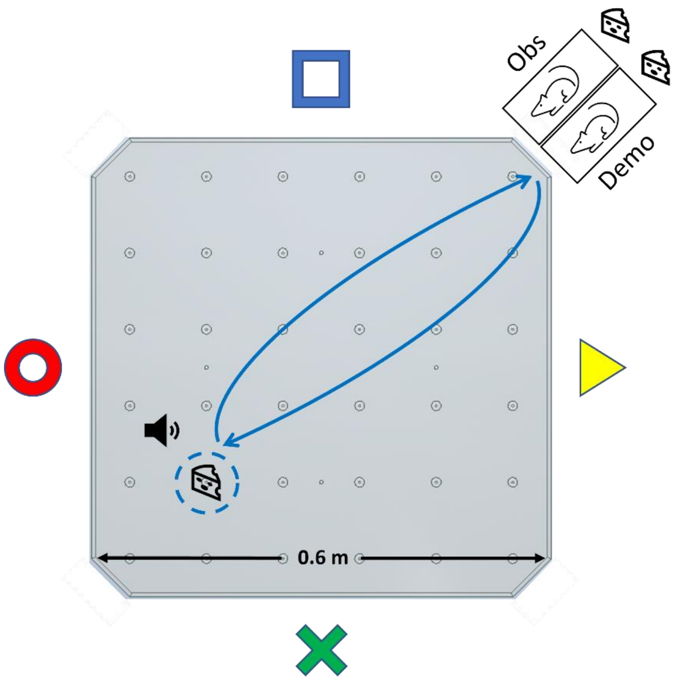


Figure 1: Experimental paradigm of the cheeseboard maze task to study spatial navigation memory.

2.3. Experimental setup

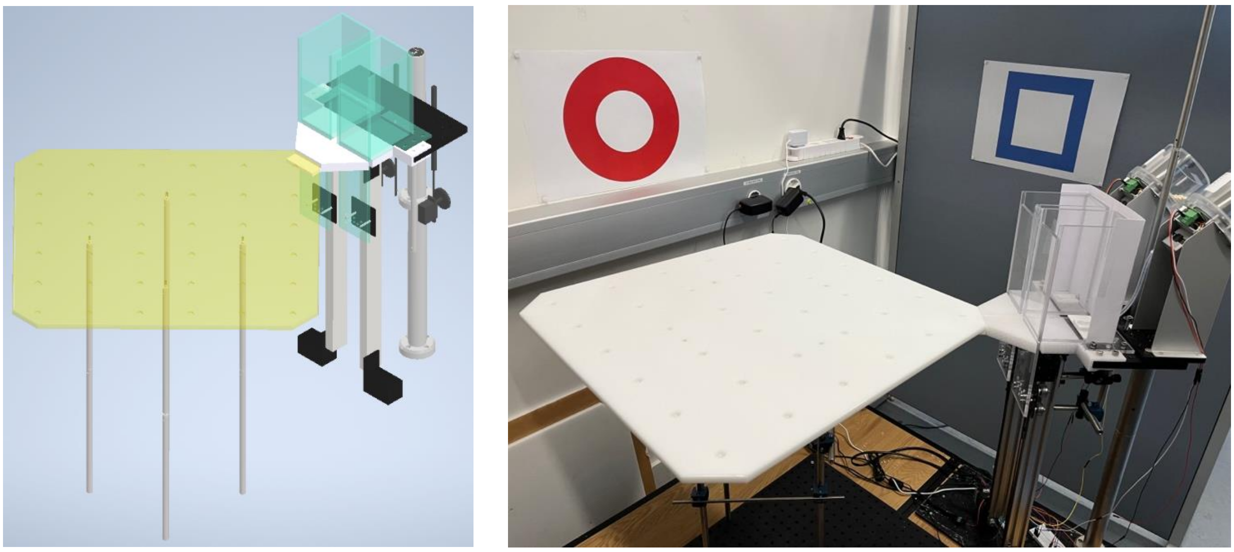


Figure 2: (left) 3D modeling diagram of the cheeseboard maze experimental setup. (right) Physical view of the cheeseboard maze setup together with the automatically controlled doors and food dispensers.

The assembly of the experimental hardware setup was designed using Autodesk Inventor, a CAD designing software (Figure 2, left). The physical representation of the experimental setup is shown in Figure 2 (right). Certain components of the platform assembly were procured from Thorlabs, while some other self-designed support and fixation parts were produced using a 3D printer (Prusa 3D). The two mouse cages are constructed from laser-cut transparent acrylic boards, with doors that can be automatically opened and closed by the belt-driven linear actuators positioned underneath. Each of the mouse cages is connected to a food dispenser (Campden Instruments) at the back, which can automatically dispense one 14mg food pellet at a time into a 3D-printed food bowl inside the cage as a food reward. Four markers with different shapes were posted in all directions to assist the mice in forming spatial memories of specific reward locations relied on the features of the surrounding environment.

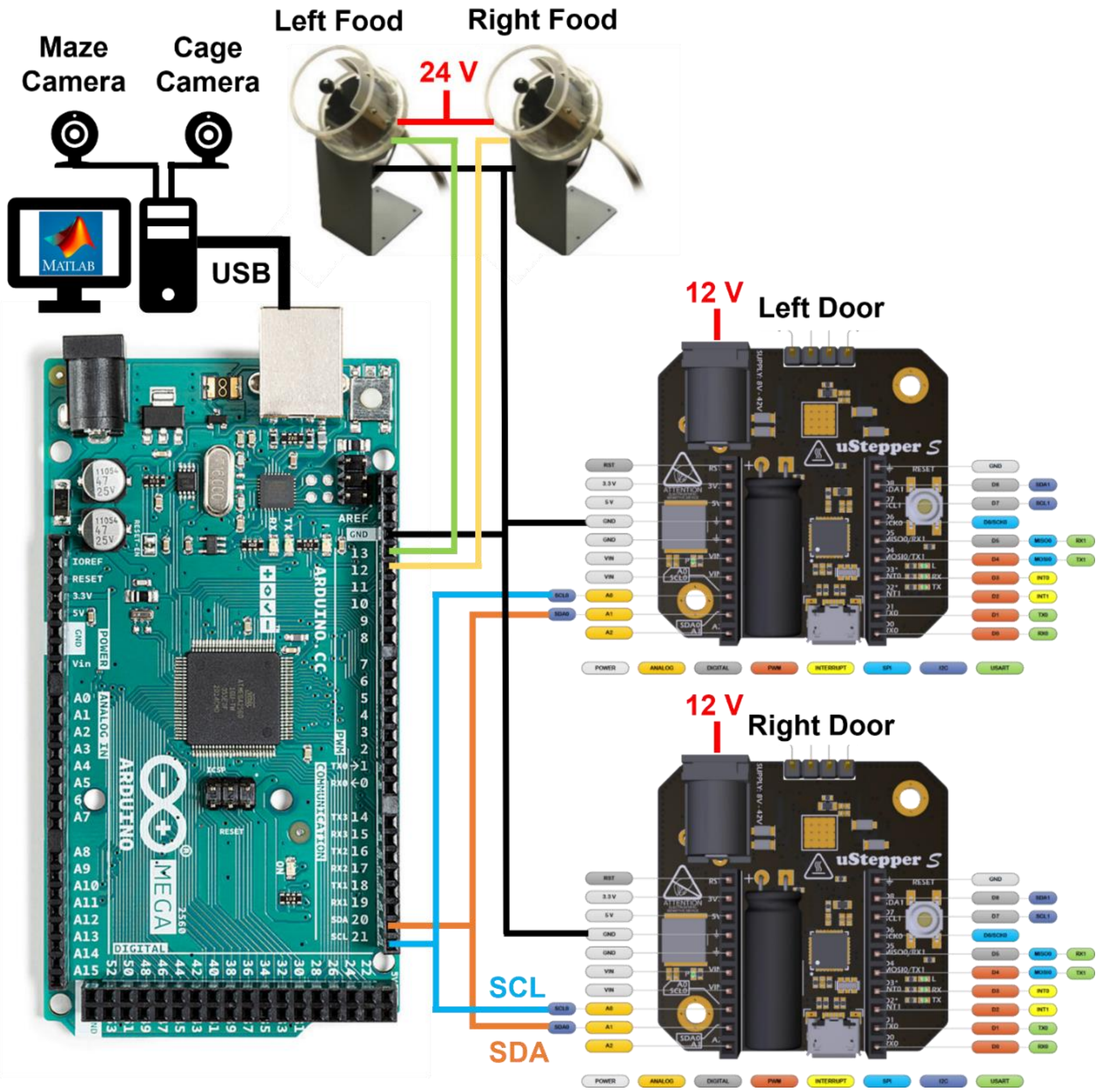


Figure 3: Connection diagram of the system. The computer equipped with MATLAB software controls two video recording cameras and the master Arduino board. The master Arduino board, which communicates with the computer via a USB port, controls the two food dispensers and communicates with the two slave uStepper boards via I2C communication. The two slave uStepper boards in turn control the doors of the animal cages.

We utilized an Arduino microcontroller as a master board to receive signal instructions from MATLAB to control the entire hardware setup (Figure 3). Specifically, the belt-driven linear actuator linked to each cage door is driven by a NEMA17 stepper motor, which is controlled by a uStepper controller board. The control code within the uStepper controller board is written using Arduino programming software. These two uStepper controller boards function as slave boards and are controlled by the master board via I2C communication. The codes in the Arduino microcontroller and the uStepper controller boards can be found in Appendix B.

2.4. Closed-loop automatic control system

The closed-loop automatic control of the task is performed with our customized MATLAB scripts. Initially, a video of the mouse exploring the maze is recorded via a USB camera (The Imaging Source Europe GmbH) mounted above the maze platform. Subsequently, a real-time tracking algorithm is developed, which involves subtracting a blank background without the mouse from the real-time captured video frame. Finally, the tracking center of the running mouse is used as condition to control cage doors and food dispensers via the microcontroller-based system. The MATLAB script of this closed-loop automatic control system can be found in Appendix A.

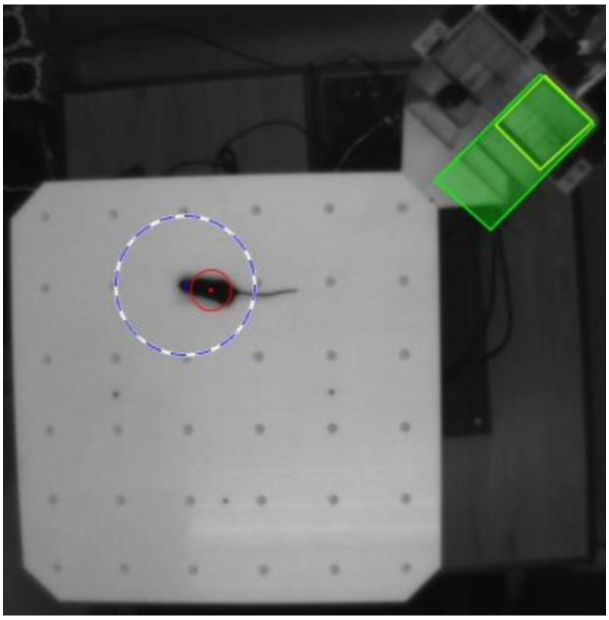


Figure 4: An example frame of the closed-loop automatic control system of spatial navigation task.

An example frame of the closed-loop automatic control system during the spatial navigation task is shown in Figure 4. For every frame we acquire in real-time, we use an image processing algorithm to track the mouse's position, which is displayed in real-time in the MATLAB figure window with the mouse's center marked by a red circle. We also draw a blue circle on the frame to indicate the selected reward zone. In addition, we draw a yellow rectangular region on the frame to represent the inner cage boundary and a green rectangular region to represent the outer cage boundary. When the mouse's position is detected within the inner cage boundary or outside the outer cage boundary, the corresponding area is filled with a semi-transparent color accordingly. To prevent the mouse from being hit by the rising cage door, the door will be closed when the mouse enters the inner boundary or exits the outer boundary.

2.5. Behavioural training protocol

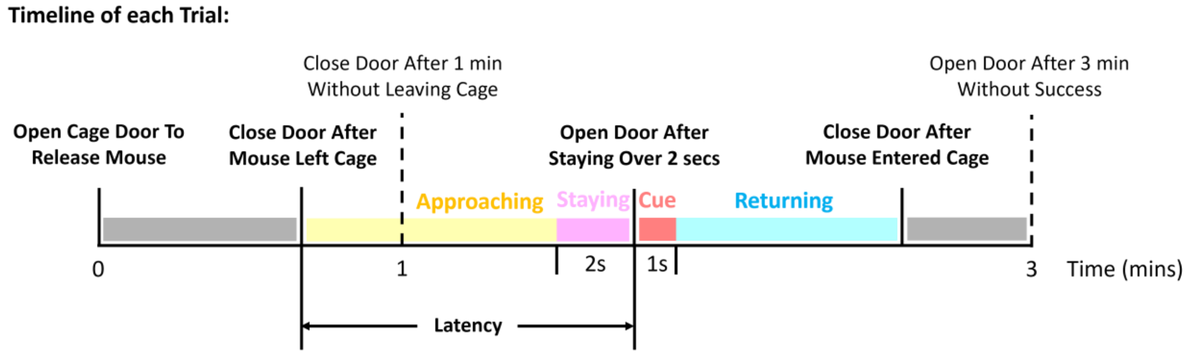


Figure 5: Timeline of one trial, including the mouse’s waiting phase inside the cage after the cage door opens, the phase of the mouse leaving the cage and approaching the target area, the phase of the mouse staying in the reward zone for 2 seconds, the phase of playing a one-second high-frequency tone as a cue after the mouse completes the task, and the phase of the mouse returning to the cage.

The timeline of one trial of the spatial learning task is depicted in Figure 5. At the start of every trial, the cage door is automatically opened to release the mouse. The tracking system then waits for the mouse to leave the cage before closing the door. If the mouse has not left the cage within one minute after the start of the trial, the cage door is closed to end the trial and then a new trial will be initiated. After the mouse leaves the cage, the first phase is the mouse approaching the reward zone. The subsequent phase involves the mouse reaching our designated reward zone, which is a circular area with a radius of 10 cm, and staying there for more than two seconds. Depending on the training conditions, a food pellet may or may not be placed in the reward zone. Upon successful completion of the two-second stay in the reward zone, we open the door to allow the mouse to return to the cage and dispense a food pellet in the cage as a reward.

Simultaneously, a high-frequency tone is played for 1 second as a cue. The final phase is the mouse returning to the cage. Once the mouse enters the cage, we close the door to end the trial. If the mouse explores the maze for more than three minutes without successfully staying in the reward zone for more than two seconds, we open the door to allow the mouse to return to the cage and classify the trial as a failed trial. We calculate the overall distance of the mouse's trajectory in four phases and the latency between the mouse leaving the cage and staying in the reward zone to quantify the mouse's performance in every trial.

To prevent the mice from using odor cues for navigation, we cleaned the maze with a 10% enzymatic animal deodorizer and cleaning solution (BIODOR GmbH) between the trials of different mice.

As illustrated in Figure 6, the basic structure of one behavioural training cohort includes the habituation session, the demo learning session, and the observational learning session.

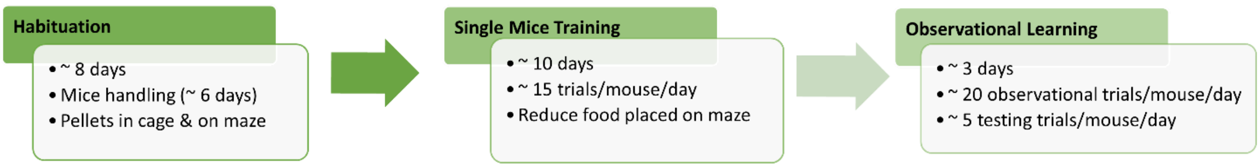


Figure 6: Flow chart of the experimental protocol for each session phase in each cohort training of mice, including the habituation session, the single mouse training session, and the observational learning session.

The purpose of the eight-day habituation session is to familiarize the experimental mice with the experimenter's operations and the structure of the cheeseboard maze behavioural task. This allows us to collect behavioural data from the mice in the experiment when they are not in a stressed or anxious state. In the first six days of the habituation session, the experimenter will handle each mouse on multiple occasions for approximately 10 minutes each day. On the seventh day of the habituation session, each mouse will be let into the cage and will receive 20 pellets dispensed in the cage consecutively in 10 minutes. This is followed by one day of habituation to the maze to better familiarize the mice with the trial structure. During this day, each mouse is released from the cage onto the maze to freely explore during three 5-minute-long trials, during which they can find 5 pellets scattered on or in the holes in each trial. All habituation procedures were performed at the same time of day as the subsequent experimental sessions.

The single mouse training session consists of approximately 10 days of behavioural training. During each day of the single mouse training session, each mouse was trained for a total duration of one hour, which approximately included 15 trials. The duration of every trial is less than 3 minutes. The inter-trial interval (ITI) between every two trials is approximately 30 seconds. In each trial, the mouse needs to complete the aforementioned behavioural task, that is, to successfully stay in the circular reward zone for a continuous two seconds. When the mouse remained in the reward zone for two seconds, a high-frequency tone was played. This was implemented to establish a conditioned association between the completion of the spatial task and the auditory cue for the mouse. To train the mice to separate the spatial memory task from reward-oriented behaviour, we gradually reduced the number of trials in which a food pellet was placed in the reward zone on the maze as an immediate reward. On the first day, a food reward was placed on the maze in every trial. On the second day of training, a food reward was placed in every other trial. This was gradually reduced to a food reward in every third trial, every fifth trial, and every eighth trial. After the shaped behavioural performance became stable, we continued the training for five more days, during which all trials had no food reward on the maze. This training paradigm is designed to shape the mice's behaviour to complete the spatial task even without the motivation of food reward on the maze.

After each individual mouse has established a stable spatial memory, they then proceed to the final observational learning session. This session lasts approximately three days, as the observer's behavioural performance will be influenced by its own exploration during the testing trials, rather than solely by the effects of observational learning. During each day's observational learning, the paired demo and observer mice are placed in two parallel transparent starting cages. In the observational trial, the demo mouse is introduced to complete the same spatial task as in the single mouse training session, serving as a demonstration. Meanwhile, the observer mouse observes the entire process of the demo mouse completing the task from within its transparent cage. In our control group, the observer's cage is surrounded by opaque cardboard to block visual observation during the observational trials, with all other conditions identical to the experimental group. During the testing trial of the observer mouse, the observer mouse is introduced to complete the spatial task, testing its learned spatial memory. In each pair of demonstrator and observer experiments, we first conduct a testing trial to assess the observer's initial performance, followed by five observational trials, and then another testing trial. This cycle repeats, with a total of approximately 20 observational trials and 5 testing trials. As with the single mouse training session, when the observer or demo mouse successfully stays in the reward zone for 2 seconds, a high-frequency tone is played.

2.6. Training conditions for each cohort

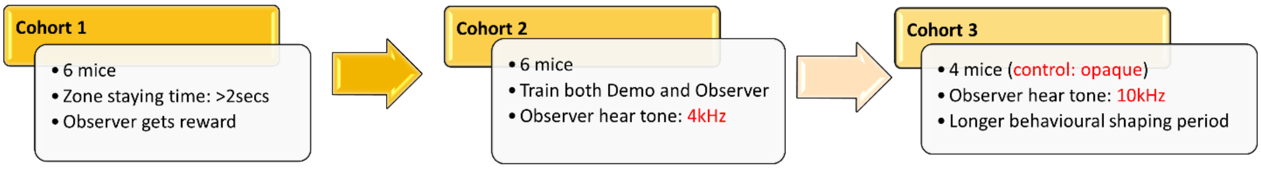


Figure 7: Flow chart of the overall procedure of the conducted mouse behavioural training experiment, including three cohorts in total.

Figure 7 summarizes the overall procedure of our conducted mouse behavioural training experiments, including three cohorts in total.

The first cohort, which included six mice, was used for a pilot experiment to test the training parameters in the protocol. In the observational learning session conducted over two consecutive days in cohort 1, which included 3 pairs of demo and observer mice, each day's experiment consisted of a total of 20 observational trials and 5 testing trials for each pair of demo and observer mice. In the observational trial, the demo mouse was introduced to complete the same spatial task as in the single mouse training session, serving as a demonstration. The demo mouse would receive a food reward both in the reward zone on the maze and back in the cage. Meanwhile, the observer mouse observed the entire process of the demo mouse completing the task from within its transparent cage. The observer mouse in the cage would receive a food reward simultaneously as the demo mouse went back into the cage. In the testing trial, the observer mouse was introduced to complete the spatial task, testing its learned spatial memory. The observer mouse would only get a food reward back in the cage after successfully completing the testing trial. In this cohort 1, the observer mice only underwent the same habituation session as the demo mice, but did not undergo the single mouse training session. That is, the observer mice were not familiar with the task structure before the observational learning session.

In the second cohort, which included six mice (three demonstrators and three observers), all six mice were trained during the single mouse training session, but the demonstrators and observers memorized different reward positions (Figure 8). This was done to familiarize the observers with the task structure, and in the subsequent observational learning phase, they needed to remap their encoded place field to the new reward zone position by observing the performance of the demonstrators. Additionally, in the following observational learning session, when the demo mouse reached the reward zone and remained there for two seconds, we played a high-frequency tone (4 kHz) instead of dispensing a food reward. This served as a cue to draw the observer mouse's attention to the performance of the demo mouse.

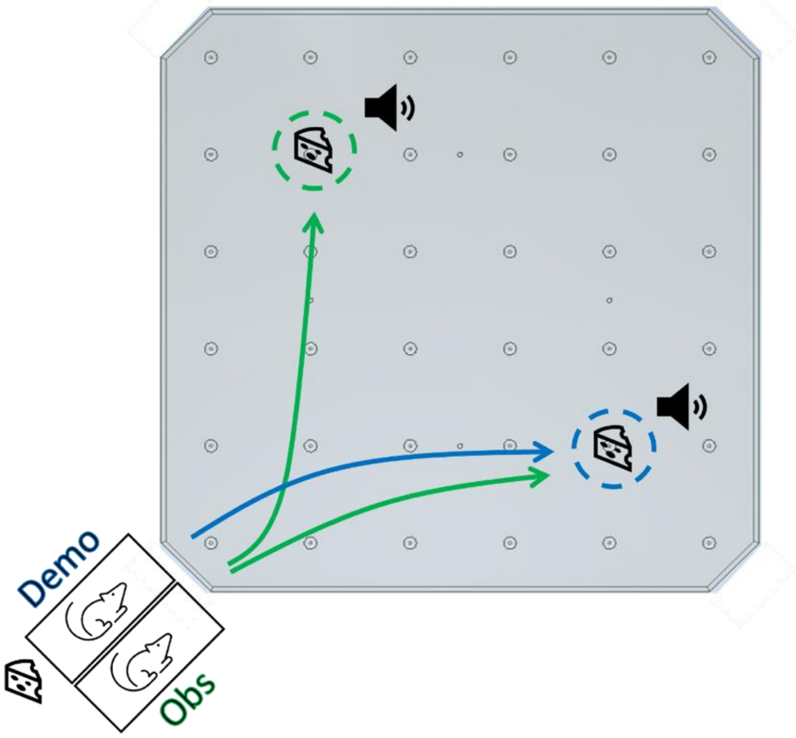


Figure 8: Experimental paradigm of the observational learning task. During the observational trial, the demo mouse demonstrates its encoded reward zone on the right side (blue area). In the testing trial, the observer mouse, relying on its observational learning experience, needs to abandon its own encoded reward zone on the left side (green area) and instead proceed to the demo mouse's reward zone on the right side (blue area) to successfully complete the trial and get the food reward back in cage.

In the third cohort, which included four mice (two demonstrators and two observers), we conducted the single mouse training session similarly to the second cohort. We trained the two demo mice with the reward position on the right side of the maze and the two observer mice with the reward position on the left side of the maze (Figure 8). Compared to the second cohort, we used a higher frequency tone (10 kHz) as a cue to attract the attention of the observer mice. Additionally, we established a control group within the third cohort, in which the observer's cage was surrounded by opaque cardboard, preventing the observer from receiving visual observational input during the observational learning trials. However, the auditory tone would still be played when the demo mouse completed the task. Furthermore, compared to the second cohort, the third cohort underwent a longer single mouse training session. During this session, we implemented a daily reduction of food placement to shape the mice's behaviour, enabling them to complete the spatial task even without the motivation of food reward on the maze.

2.7. Analysis of behavioural data

2.7.1. Learning performance analysis

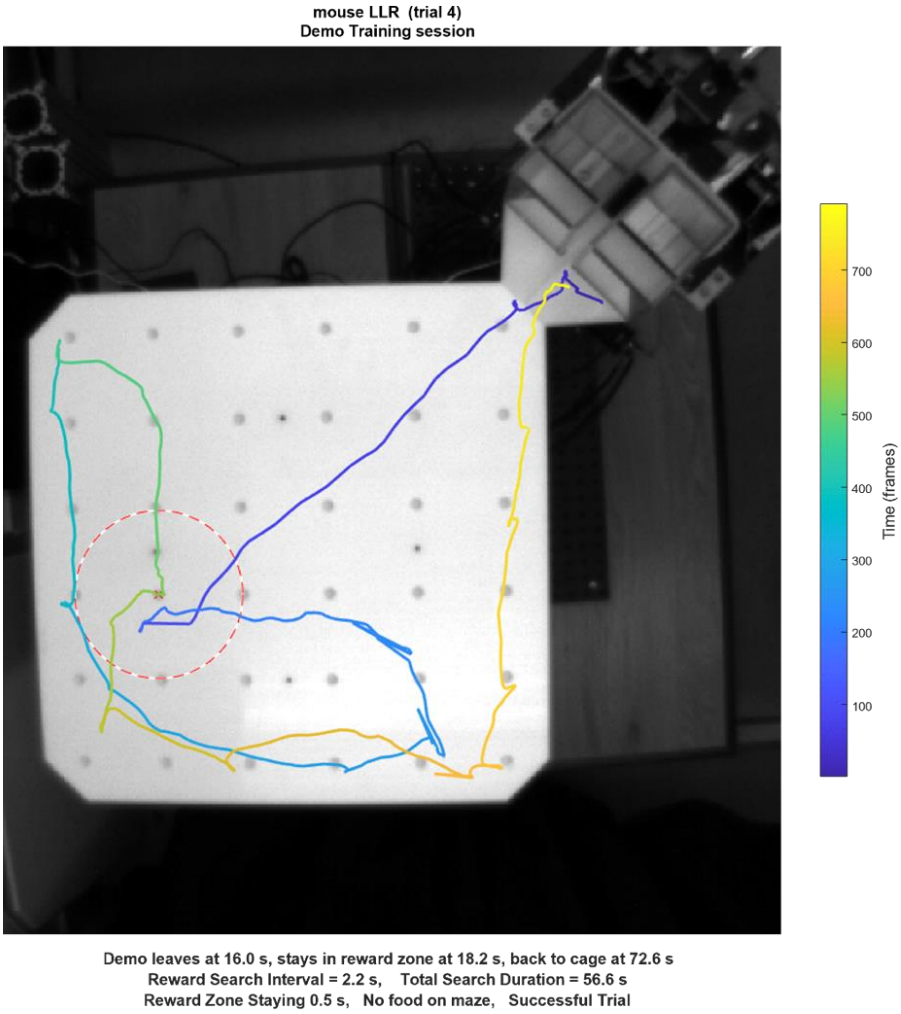


Figure 9: An example trajectory plot of a demo mouse's tracked position in one trial. The colorbar Indicates time sequence.

During both the single mouse training session and the observational learning session, we calculated the trajectory distance of the mice completing the cheeseboard maze spatial task to quantify their learning performance. Figure 9 shows a complete trajectory from the start of a trial, when the mouse leaves the cage, to the end of the trial when the mouse returns to the cage, obtained by the real-time tracking of the mouse's position in an example trial. We obtained the trajectory distance for each trial by summing the framewise calculated distances of the mouse's position changes. In addition, during the single mouse training session, we also calculated the latency from when the mouse leaves the cage to when it successfully reaches the reward zone and stays for two seconds. The timepoints to calculate the latency were also recorded by the real-time tracking system. The latency serves as another measure to quantify the mice's learning performance.

The custom MATLAB scripts used to perform analysis for all raw data are available at https://github.com/dduyh/Reward-directed_Spatial_Observational_Learning_in_Mice. Some following statistical t-test analyses were done in GraphPad Prism 9.5.

2.7.2. Head direction analysis

To analyze the behaviour of the observer mice during observational learning, we sought to quantify the head direction of the observer mice to characterize their attention state towards the demo mice's behaviour. We first used the DeepLabCut software to annotate the positions of the snout and both ears of the observer mouse in each frame of the video recorded during the observational learning period (Figure 10).

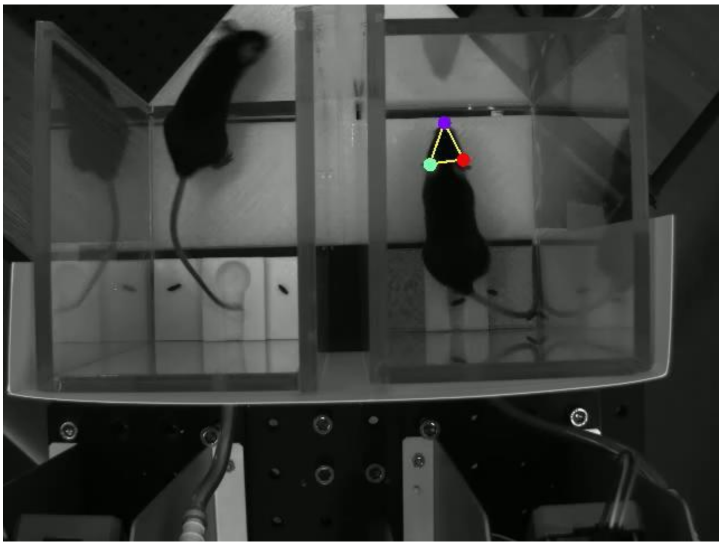


Figure 10: An example frame of the behavioural video of two mice in cages during the observational learning. The head directions of the observer mice in the right cage are tracked with DeepLabCut software.

Subsequently, we used custom MATLAB scripts to calculate the angle of the head direction vector, which is the vector from the midpoint of the two ears to the snout, relative to the horizontal direction. To be consistent with the origin of the coordinates in the upper left corner of the video annotated by the DeepLabCut software, we set the polar coordinate direction for calculating the head direction angle to be clockwise for better subsequent visualization (Figure 11, top). The bottom part of Figure 11 shows two example figures of the subsequent head direction analyses, which are the timeseries of head directions during different behavioural phases (Figure 5) of one trial (Figure 11, bottom left), and the distribution of head directions for that trial (Figure 11, bottom right). A head direction around -90° indicates that the mouse's head is facing towards the maze, while a head direction around 90° suggests that the mouse's head is facing towards the food dispenser outlet located behind the cage.

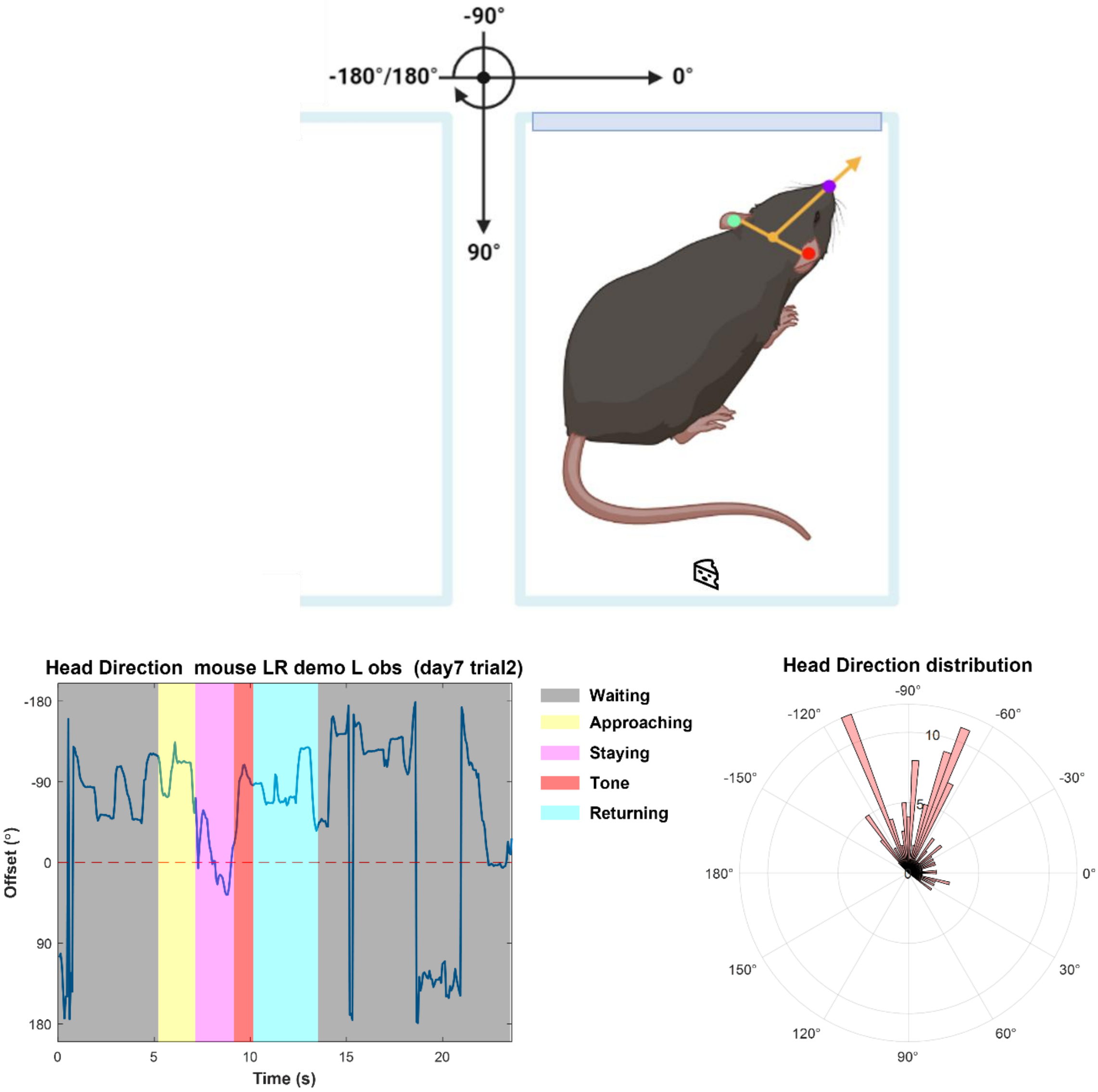


Figure 11: (top) A schematic illustrating the analysis procedure of head directions of the observer mouse. (bottom left) The timeseries of head directions during different behavioural phases (waiting, approaching, staying, tone playing, returning) of one trial. (bottom right) A rose plot showing the distribution of head directions for the trial on the left.

3. Results

3.1. Learning performance of single mouse spatial navigation

3.1.1. Trajectory distance used to quantify learning performance

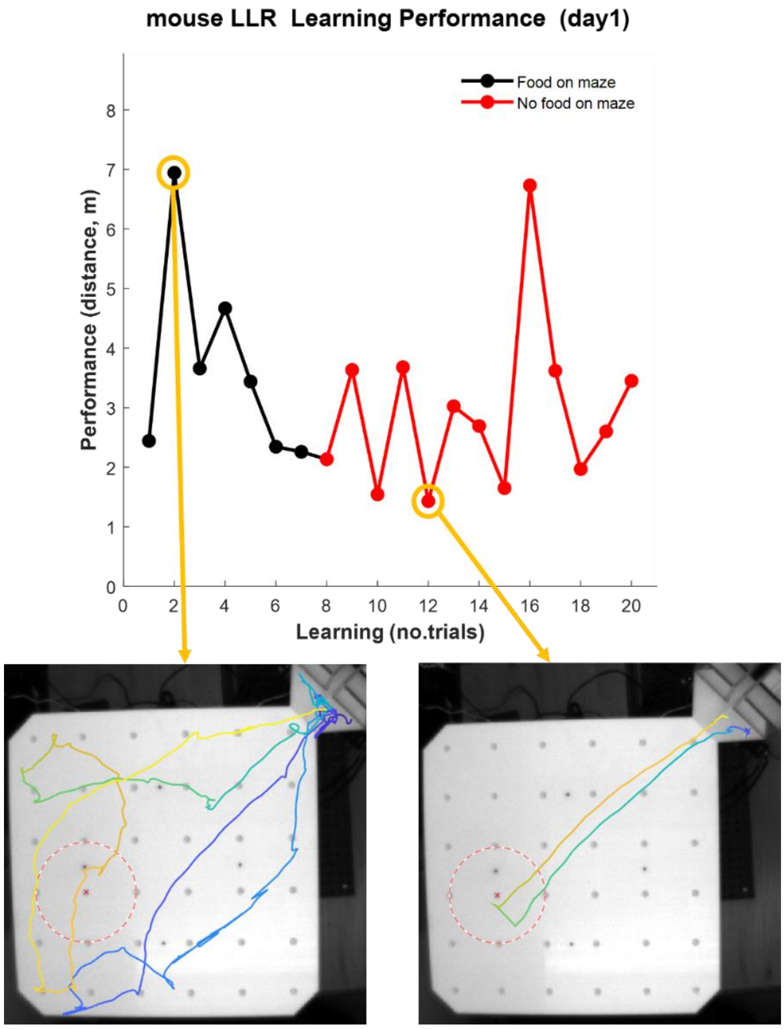


Figure 12: (top) An example learning performance curve of one mouse from cohort 1 that is quantified by the distance travelled to find the reward per trial. (bottom) A mouse's trajectories travelled to the reward zone in an early trial and a late trial.

The cohort 1 was used for a pilot experiment to test the training parameters in the protocol. By observing the performance of mice executing the spatial learning task on the cheeseboard maze in cohort 1, we found that during the initial few trials, when the mice were unfamiliar with the goal-oriented spatial task, the distance of their trajectories was relatively long. This is due to the exploratory nature of their movements as they randomly navigate the unfamiliar environment. However, once the mice understood the goal-oriented spatial task, they would more directly approach the reward zone and quickly return to the cage after completing the task. As a result, the distance of their trajectories became shorter. Therefore, the distance of the trajectory can be used as a metric to quantify the learning performance of the mice (Figure 12). Additionally, based on the results of the learning performance across several attempts under different conditions in cohort 1, we ultimately set the threshold that mice are required to stay in the reward zone for two seconds to complete a successful trial and receive a food reward back in the cage. The rationale behind the two-second threshold is as follows: If the threshold is set too short, mice might frequently just pass by the reward zone accidentally without consciously forming a spatial memory. Conversely, if the threshold is set too long, mice might struggle to stay in the reward zone for the required duration, making task completion challenging. Thus, two seconds is a reasonable threshold.

3.1.2. Rapid improvement in memorizing reward location

In our pilot trials in cohort 1, we preliminarily decide the behavioral training parameters. Starting from cohort 2, we implemented a standardized group behavioral training and conducted a complete statistical analysis to quantify the mice's learning performance (Figure 13). In the single mouse training session of cohort 2, we consistently placed a food pellet in the reward zone for every trial of the 6 trained mice over the first five training days. During each day of the single mouse training session in cohort 2, each mouse was trained for approximately 10 trials (Figure 13, top). We observed a notable consistency in the performance of the six mice during this training period. The mice's ability to remember and recall the position of the food rewards improved rapidly, reaching a very stable behavioural paradigm after just a few days of training (Figure 13, bottom). This suggests that when a food reward on the maze is used as a motivational trigger, the mice are capable of encoding a long-term spatial memory of the reward position, and their behaviour can be effectively shaped.

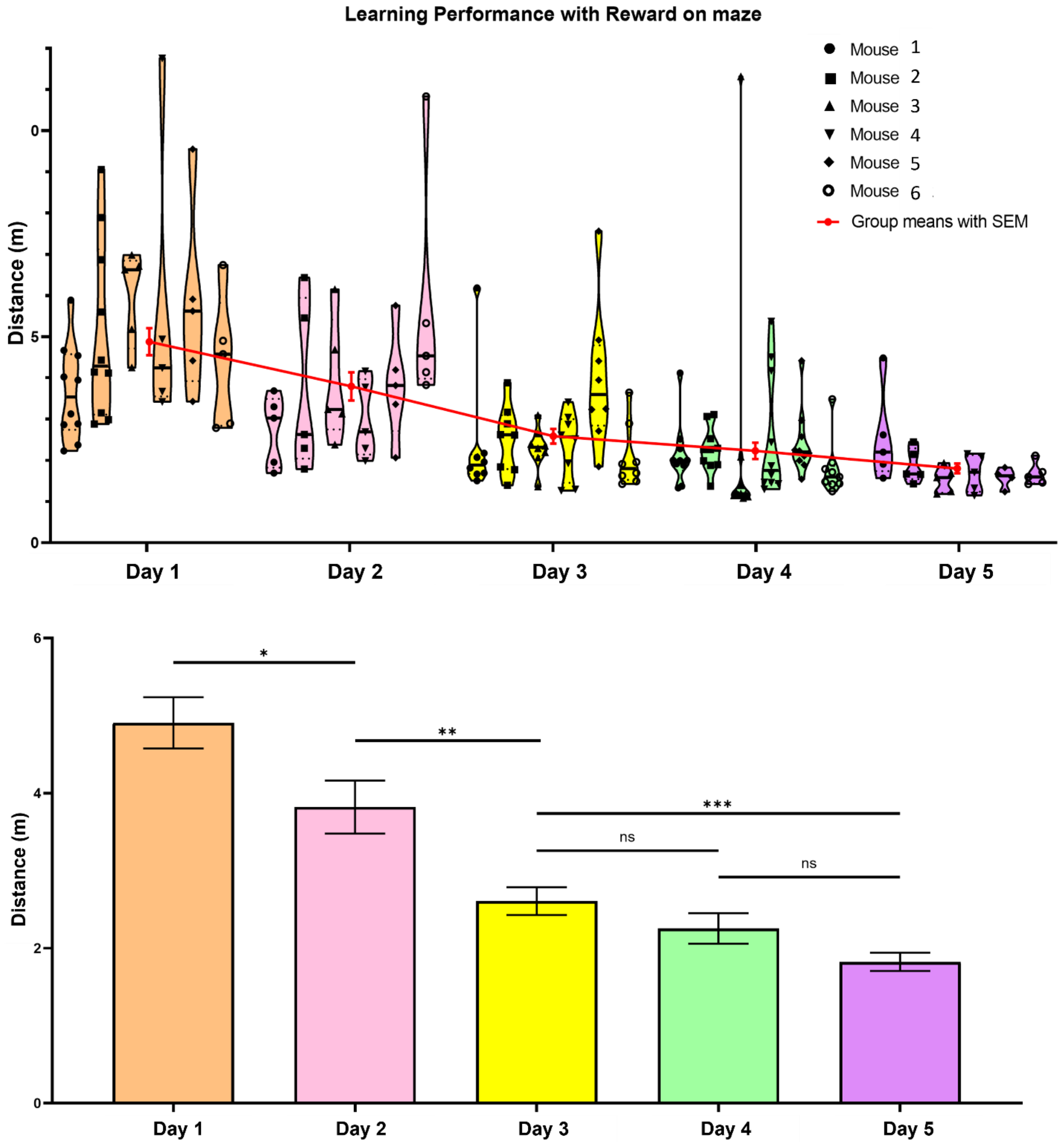


Figure 13: (top) A complete violin plot depicting the learning performance, as measured by trajectory distance, of 6 mice trained in cohort 2 with reward placed on the maze (medians with quartiles indicated). (bottom) The bar plot integrates the daily learning performance of the 6 mice from the upper figure (means ± s.e.m.), and calculates unpaired t-tests with Welch's correction between every two-day pairwise comparisons. Statistically significant differences in the Welch-corrected comparisons are indicated by asterisks, with * = p < .05, ** = p < .01, *** = p < .001, and ns = not significantly different.

3.1.3. Latency to reward used to quantify learning performance

Given that the function of cheeseboard maze task is similar to the Morris water maze, both serving as behavioural tasks to study spatial learning and memory capabilities, we drew inspiration of the analysis methods from the Morris water maze, which is a classical and widely used behavioural task. This reference indicated that in addition to the cumulative distance of the animal's trajectory, the latency to reach the goal position is also a commonly used metric to quantify a mouse's learning performance. Consequently, in addition to using the trajectory distance to quantify the learning performance of the mice, we also calculated the latency, defined as the time from when the mouse leaves the cage to when it successfully reaches the reward zone and stays for two seconds. This type of data is known as 'time-to-event' data, which captures the time until an event occurs. If the event did not occur, which in our case meant that the mouse did not find the food reward during the 3-minute exploration time, these datapoints were censored, as the true time until the event was unknown.

The time-to-event data, or latency, can be analyzed using a statistical approach called survival analysis. This method can handle censored data, which are datapoints that are missing because the event had not been observed during the sampling time. In the survival analysis, the time-to-event data can be displayed as a step function to show the cumulative probability that an event happened after each timepoint. The Kaplan-Meier approach can be used to estimate these survival curves non-parametrically and account for censored observations.

Our dataset for the survival analysis in cohort 2 consists of 206 datapoints with no censored datapoints, spread across 5 survival curves for each day's training. Each datapoint represents the time it took a subject to visit a given reward location and stay for two seconds. The Kaplan-Meier survival curves in Figure 14 (left) show the estimated cumulative probability that the event (successfully approaching and staying at a reward position) has occurred by each timepoint. We observed a shift in the probability distribution towards shorter latency durations as the days of training progressed. This observation aligns with the conclusion drawn from the trajectory distance quantification. That is, the mice's capability to remember and approach the reward zone enhanced rapidly, reaching a very stable behavioural paradigm after only a few days of training (Figure 14).

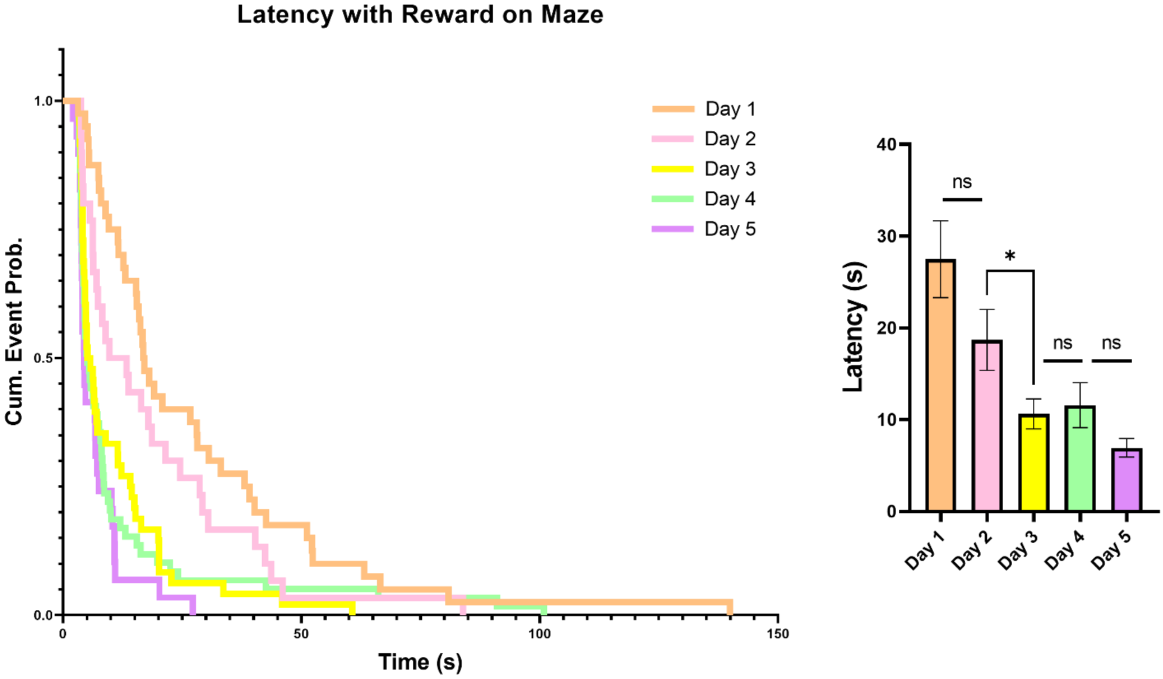


Figure 14: (left) The Kaplan-Meier survival curves depict the cumulative probability of successful completion of approaching and staying at a reward location by each timepoint in every trial for all the 6 mice over the 5-day training period. (right) The bar plot integrates the daily learning performance of the 6 mice from the left figure (means ± s.e.m.), and calculates unpaired t-tests with Welch's correction between every two-day pairwise comparisons. Statistically significant differences in the Welch-corrected comparisons are indicated by asterisks, with * = $p < .05$ and ns = not significantly different.

Drawing from various methods used to analyze rodent behavioural data in the Morris Water Maze (36), we expanded our analysis beyond the commonly calculated metrics of cumulative distance of trajectory and latency. We also examined the trajectory patterns to study the exploration strategies that mice employed during the spatial task. Observing the distribution of trajectories in cohort 2 (Figure 15), we noted that on the initial day of training, when the mice were unfamiliar with the task, they randomly explored everywhere on the maze. However, as the training days progressed, the mice learned the goal-oriented task better and began navigating directly towards the reward zone.

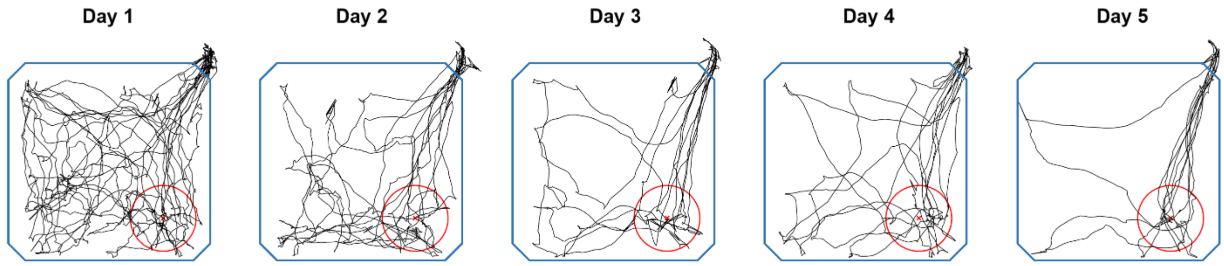


Figure 15: Trajectory plots of every day's first 5 trials across 5 consecutive training days for a mouse in Cohort 2.

3.1.4. Behavioural shaping with daily reduction of food placement

The preliminary training results from cohort 1 and cohort 2 revealed that when a food reward on the maze served as a motivational trigger, the mice's behaviour could be rapidly trained to directly complete the spatial learning task. Indeed, this type of learning capability aligns with findings reported in previous studies (21). Subsequently, considering that the observer in the subsequent observational learning session was exposed to a purely spatial task, in cohort 3, we aimed to separate the spatial learning from the reward-oriented learning. The intention was that the mice could still remember the spatial location even without food placed in the reward zone (8). In other words, the mice were required to complete the spatial behavioural task, similar to a lever pressing task, to receive a reward back in the cage.

To shape the mice's behaviour, we gradually reduced the food placement in the reward zone on the maze. Specifically, on the first day, a food reward was placed on the maze in every trial. On the second day of training, a food reward was placed in every other trial. This was gradually reduced to a food reward in every third trial, every fifth trial, and every eighth trial. In the trials without food on the maze, the mice needed to rely on the delayed rewards to shape their behaviour to go to and stay at a specific encoded location for 2 seconds.

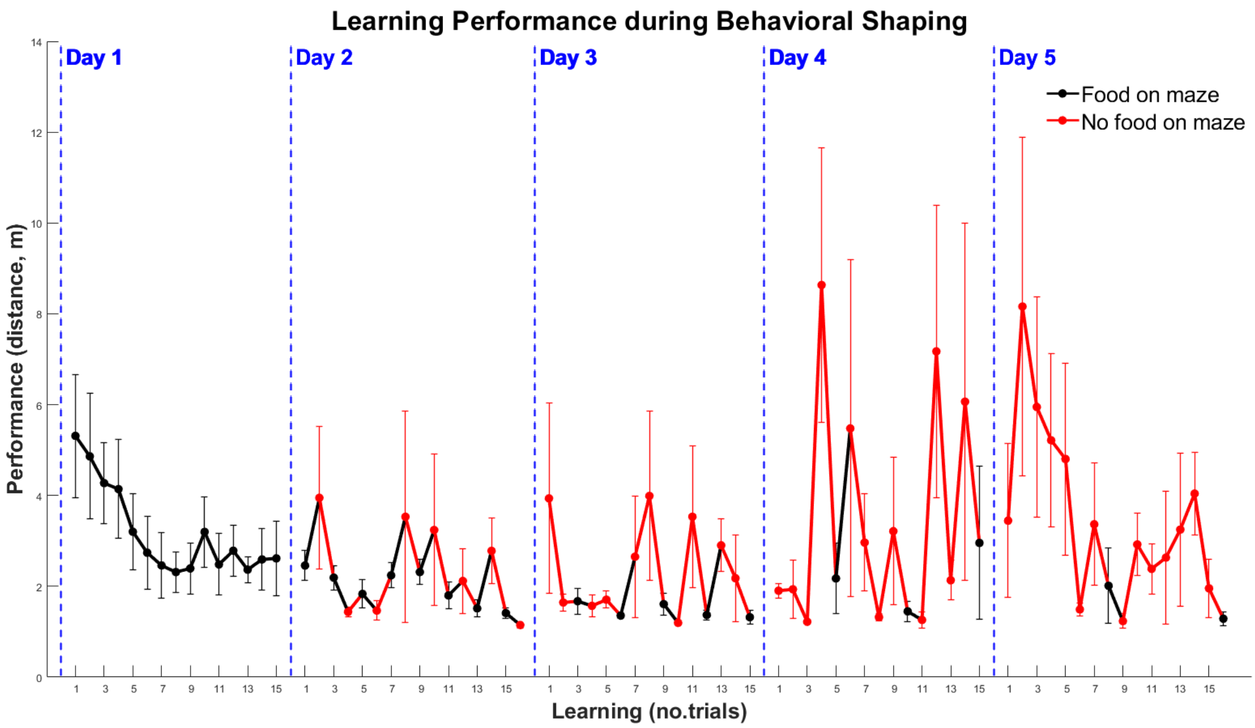


Figure 16: Averaged learning performance of the 4 mice in cohort 3 measured by trajectory distance during the 5-day behavioural shaping with daily reduction of food placement on the maze. Black dots represent trials with food reward on the maze for the 4 mice, while red dots represent trials without food reward on the maze for the 4 mice. The error bars represent the standard error of the mean (s.e.m.) for the 4 mice.

The results of this behavioural shaping paradigm on the 4 mice trained in cohort 3 showed that on the first day of training, when a food reward was placed on the maze in every trial, the mice quickly formed a stable behaviour of directly approaching and returning from the reward zone. On the second and third days of training, when only some trials contained an immediate reward on the maze, the mice still performed well in remembering the reward location in most of the trials. However, on the fourth day of training, when most trials lacked a food reward on the maze, there was a significant increase in the variance of the mice's performance. In the subsequent day of training, the mice's performance again progressively became stable (Figure 16).

3.1.5. Spatial memory recall triggered by non-immediate reward

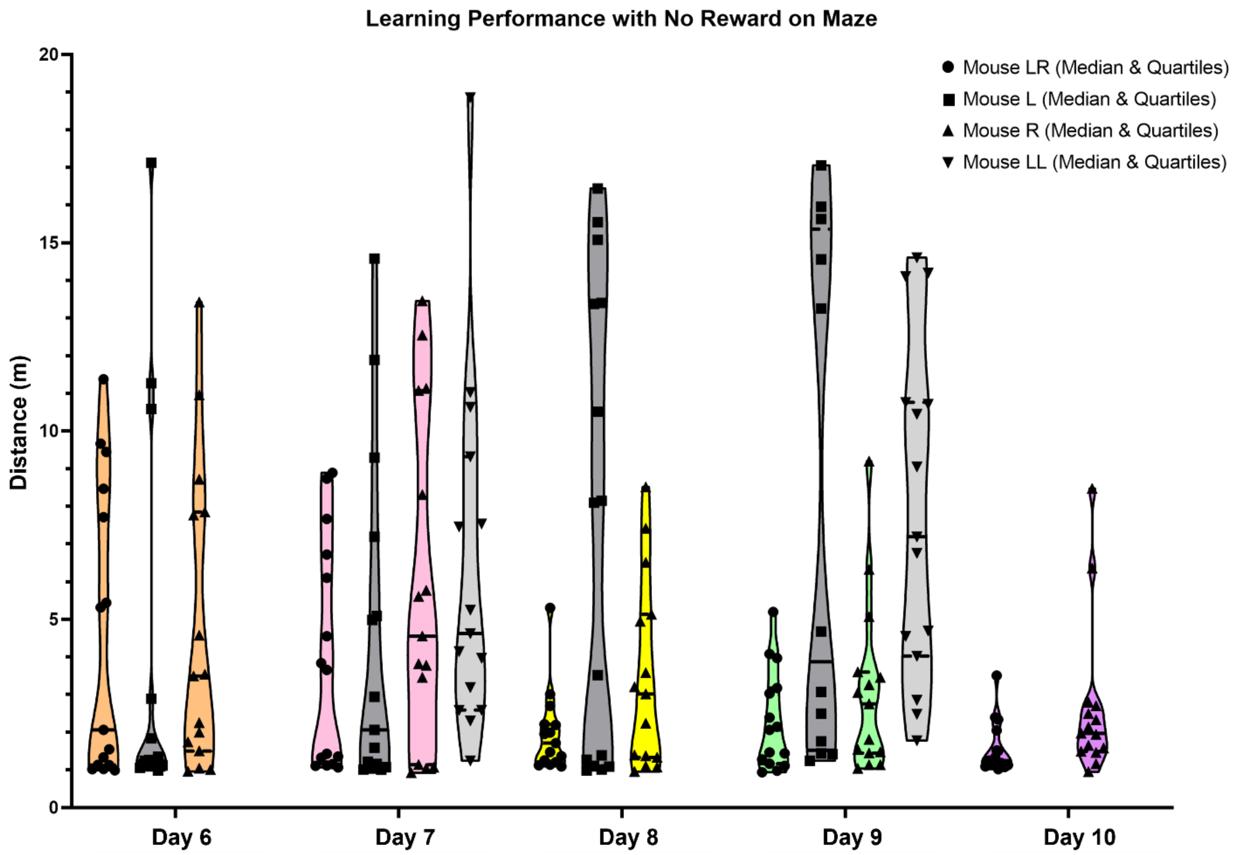


Figure 17: A complete violin plot depicting the learning performance, as measured by trajectory distance, of 4 mice trained in cohort 3 without reward on the maze (medians with quartiles indicated). The two mice with better performance are highlighted in color, while the two mice with less optimal performance are marked in gray.

After the initial five days of behavioural shaping in cohort 3, once the performance of the 4 mice in the spatial memory retrieval task had become relatively stable, we proceeded to completely remove the food reward on the maze. This was done with the aim of training the mice to complete the spatial memory retrieval task and obtain a delayed reward in the cage, even in the absence of an immediate reward trigger on the maze. Figure 17 shows the behavioural performance, quantified by trajectory distance, of the 4 mice trained in cohort 3 under conditions without a food reward trigger on the maze.

In Figure 17, some mice do not have complete five-day training data. This is because when there is no food reward on the maze to encourage the mice to actively start exploring the spatial task, the motivation of some mice significantly decreases, as evidenced by their reduced activity to leave the cage. If a mouse does not voluntarily leave the cage to start the task within one minute after the start of the trial, the cage door is automatically closed to end the trial, and then a new trial is initiated to replace the previous failed trial. The missing mice data in Figure 17 is due to the fact that the mice essentially stayed in the cage and were unwilling to come out to complete the task during the training process of the day, leading to the cessation of training for that day. We found that there is an inherent variance in the performance of individual mouse. In the experimental cohort 3 we conducted, we prioritized training the mice with higher motivation to enter the subsequent observational learning session, while the mice that did not show particularly motivated behaviour in the absence of immediate food reward might also eventually achieve good performance if we extend the training time. This suggests setting performance-based milestones for this single mouse training session, rather than relying on a fixed number of training days. This approach ensures that mice have adequately learned before progressing to the next observational learning session.

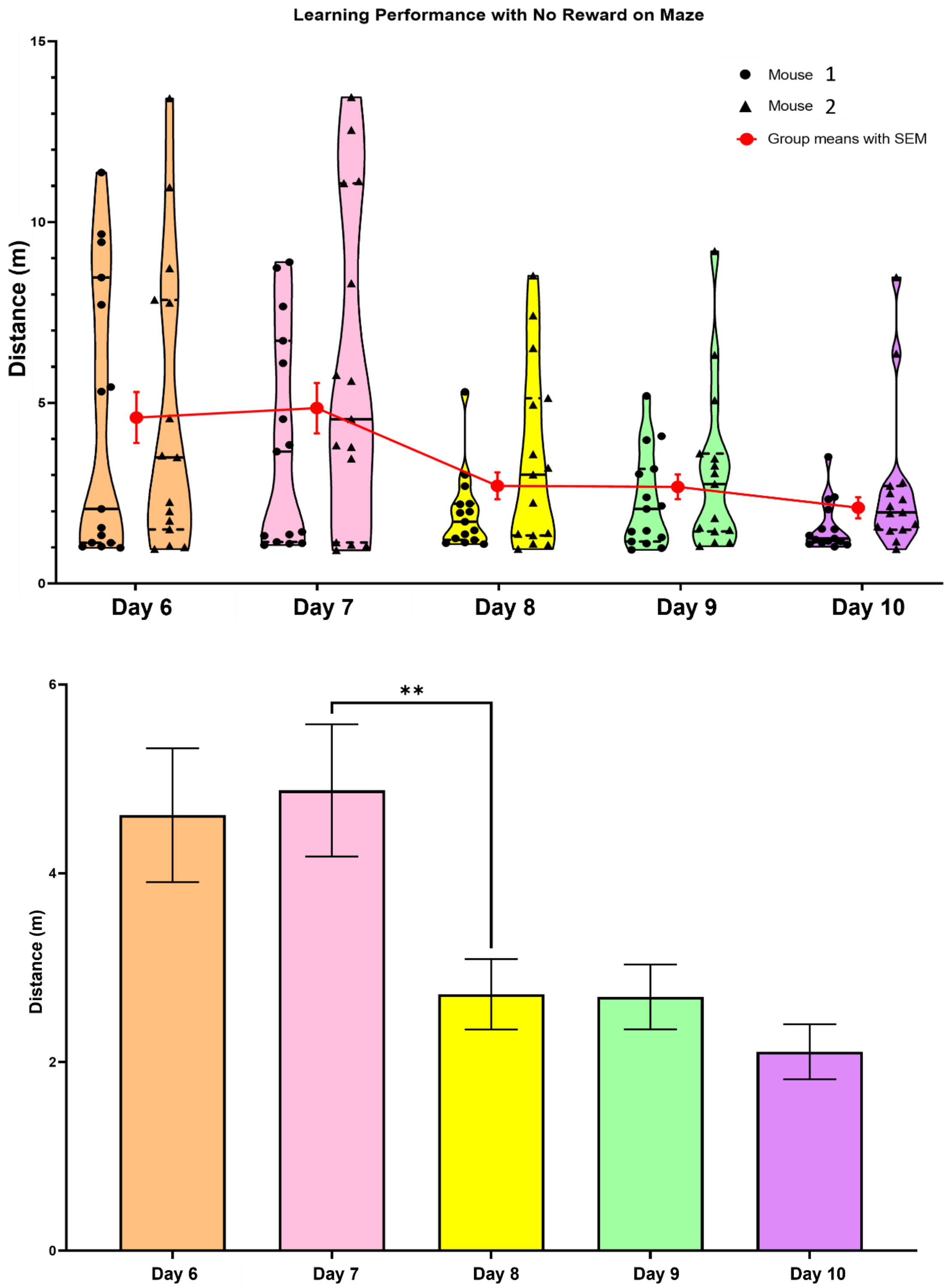


Figure 18: (top) A violin plot depicting the learning performance, as measured by trajectory distance, of the 2 well-performed mice trained in cohort 3 without reward on the maze (medians with quartiles indicated). (bottom) The bar plot integrates the daily learning performance of the 2 mice from the upper figure (means ± s.e.m.), and calculates unpaired t-tests with Welch's correction between every two-day pairwise comparisons. Statistically significant differences in the Welch-corrected comparisons are indicated by asterisks, with ** = p < .01.

Overall, from the results of this subsequent continuous five-day training in cohort 3, we found that when all trials lacked food rewards, a subset of mice (2 out of 4) were still able to accurately complete the spatial behavioural task after a few days of training. The learning performance, quantified by trajectory distance, of the two better-performing mice on the tenth day (Figure 18) of training in cohort 3 was essentially the same as the training results on the fifth day in cohort 3 (Figure 13). This suggests that they achieved a performance under the condition of no food reward on the maze that was nearly as good as when there was a food reward on the maze.

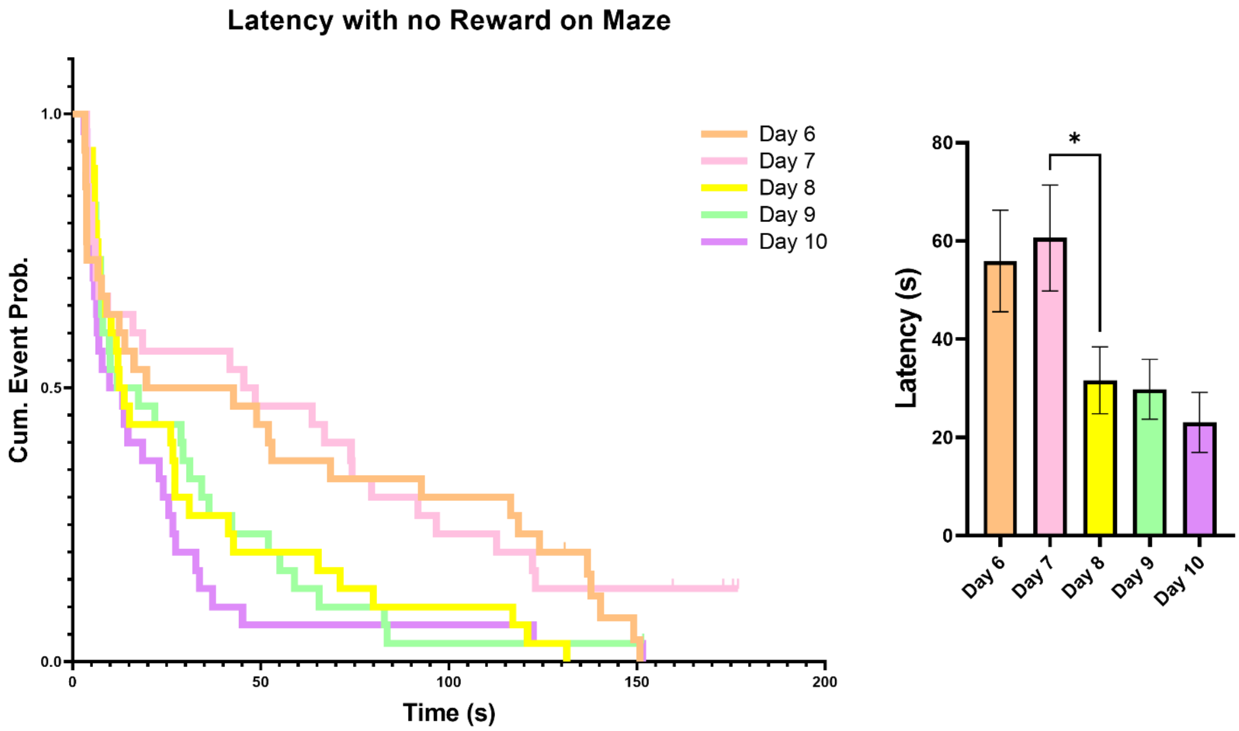


Figure 19: (left) The Kaplan-Meier survival curves depict the cumulative probability of successful completion of approaching and staying at a reward location by each timepoint in every trial for the 2 well-performed mice over the 5-day behavioural shaping period. (right) The bar plot integrates the daily learning performance of the 2 mice from the left figure (means ± s.e.m.), and calculates unpaired t-tests with Welch's correction between every two-day pairwise comparisons. Statistically significant differences in the Welch-corrected comparisons are indicated by asterisks, with * = $p < .05$.

For the subsequent five days of training without food reward on the maze, we also conducted a survival analysis on the latency of each trial completed by the 2 well-performing mice. Our dataset for the survival analysis in cohort 3 consists of 150 datapoints with 6 censored datapoints, spread across 5 survival curves for each day's training. For the 6 censored datapoints, the time variable reflects the total time (around 3 minutes) the subject had spent searching on the maze.

Similar to the results in Figure 14, we observed a shift in the probability distribution towards shorter latency durations as the days of training progressed.

However, when comparing with the training results with food reward on the maze in Figure 13 and Figure 14, we found that when the behavioural performance reached stability without food on the maze, although the learning performance quantified by trajectory distance was basically consistent with the stable performance achieved when there was food on the maze (Figure 13, bottom; Figure 18, bottom), the learning performance quantified by latency showed that the average latency of the final stable performance of the mice was 23 seconds (Figure 19, right), which is much longer than the average of 7 seconds when there was food on the maze (Figure 14, right). This may be due to the fact that in the absence of food on the maze, the mice showed less motivation in the process of approaching the reward zone, resulting in a slower speed compared to the situation when there was food on the maze.

3.2. Observational learning performance

3.2.1. Learning performance of demonstrator and observer mice

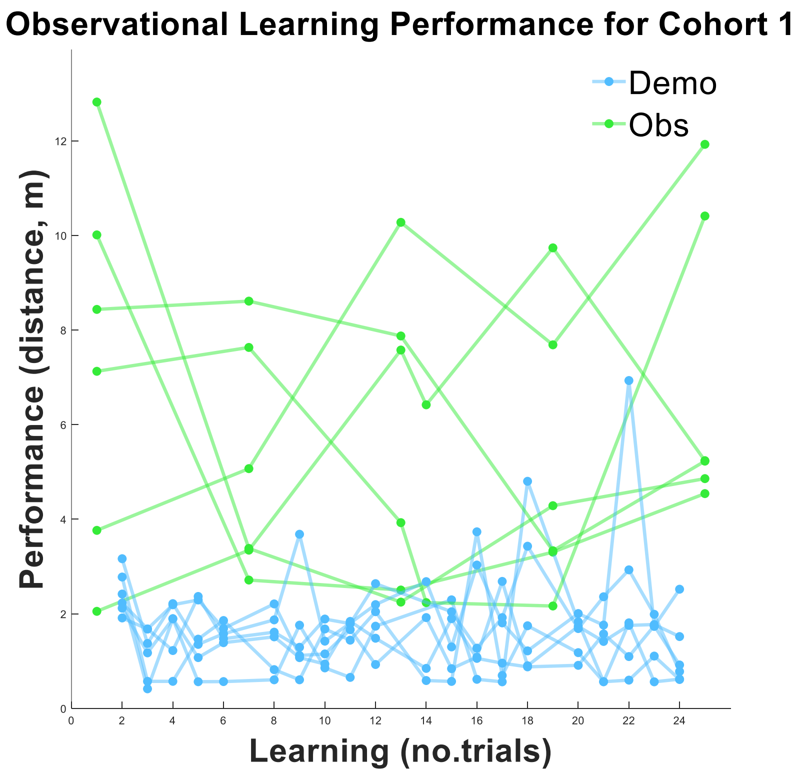


Figure 20: Learning performance curves of observational learning in cohort 1, including the learning performance curves for 3 pairs of demo mice (blue) and observer mice (green) across 2 consecutive days during the observational learning sessions.

As depicted in Figure 20, which illustrates the observational learning performance in cohort 1, the demo mice, having been through the single mouse training session and receiving a food reward in the reward zone on the maze during the observational trial, exhibited relatively stable and consistent performance (blue lines in Figure 20). Conversely, the observer mice, which performed one testing trial after observing the demo mice perform five trials, exhibited a larger variance in performance (green lines in Figure 20). Among the six performance curves of the observer's testing trials, three showed a relatively noticeable improvement through the number of observational trials. This larger variance in observer's performance may be attributed to the observer mice's lack of prior experience with the spatial task structure, coupled with the absence of food reward in the reward zone on the maze as a motivational trigger during the testing trials.

3.2.2. Head direction analysis characterizes observer's attention

In addition to calculating the trajectory distance to quantify the observational learning performance, we also tracked the body parts of mice and then calculated the head directions to characterize the attention dynamics of the observer mice during the observational learning session. The top part of Figure 21 presents the timeseries of head directions of the observer mice during a total of 6 seconds, which includes the last 3 seconds of the demo mice returning to the cage after completing the spatial task, the 2 seconds after the demo mice entered the cage and a food pellet was dispensed to the observer mice, and the 1 second following that, across a total of 59 observational trials in cohort 1.

From the head direction analysis of observational learning in cohort 1, we found that when the observer mouse in the cage received a food reward simultaneously as the demo mouse completed the task, the observer mouse would essentially just wait for the food pellet to be dispensed and not pay attention to the demo mouse's behaviour. The bottom left part of Figure 21 shows the head direction timeseries for 39 out of the 59 observational trials in cohort 1, where the head direction of the observer mice was essentially around 90°, indicating that the observer mouse's head was consistently facing towards the food dispenser outlet located behind the cage during the 6 seconds before and after the food pellet was dispensed. We referred to this type of attention as 'sustained attention to reward'. The remaining 20 trials with less stable head direction timeseries were categorized as 'inattentiveness'. The bottom right part of Figure 21 shows the distribution of these two types of attention in cohort 1.

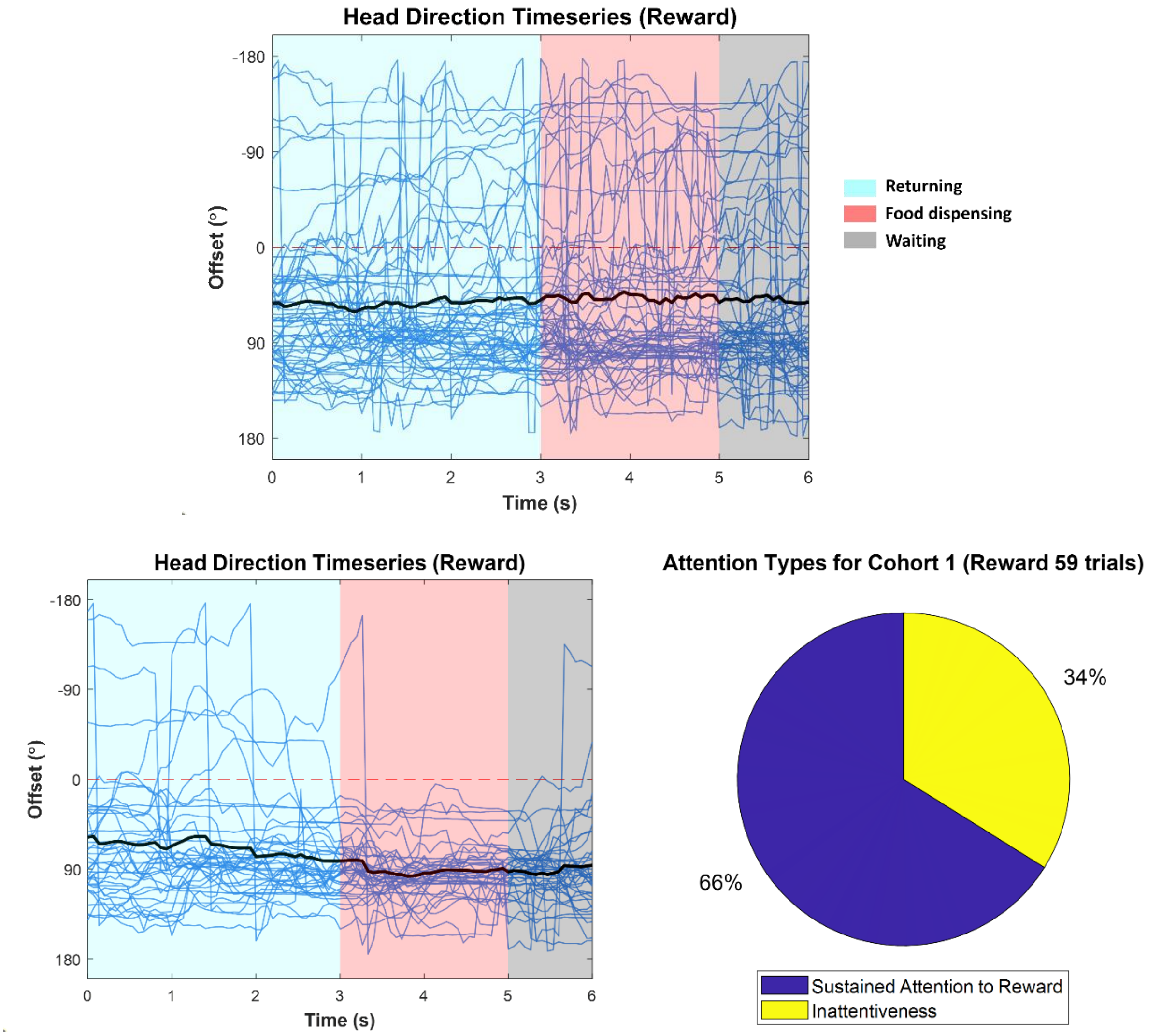


Figure 21: (top) The timeseries of head direction of the observer mice during a total of 6 seconds, including 3 phases, across a total of 59 observational trials in cohort 1. Blue lines represent the head direction timeseries for observer mice in individual trial, while the black line represents the average of head direction angles across all trials. (bottom left) Head direction curves of 39 trials which show sustained attention to the food dispenser outlet located behind the cage. (bottom right) The distribution of the two types of attention in cohort 1.

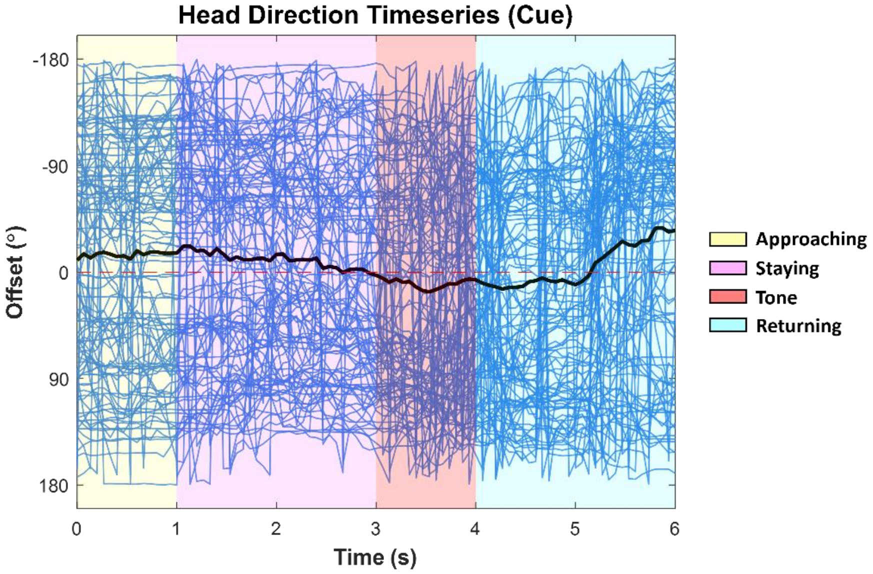
3.2.3. Auditory cue can better attract the observer's attention

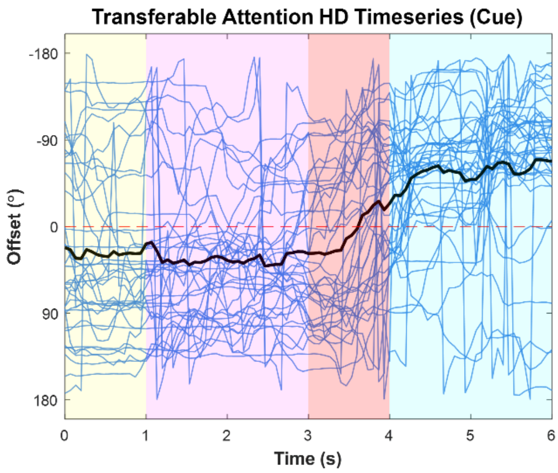
The results from the cohort 1 experiment indicated that when the observer mouse in the cage received a food reward simultaneously as the demo mouse completed the task, the observer mouse would primarily await its food reward and neglected the demo mouse's behaviours. Recognizing that this simultaneous reward protocol didn't effectively promote attentive observation of the spatial task by the observer mice, we adjusted our methodology from cohort 2 onwards. In cohort 2, when the demo mouse successfully stayed in the circular reward zone for a continuous two seconds during the observational trials, we played a high-frequency tone instead of dispensing a food pellet to the observer. The tone served as a cue to draw the observer mouse's attention to the performance of the demo mouse.

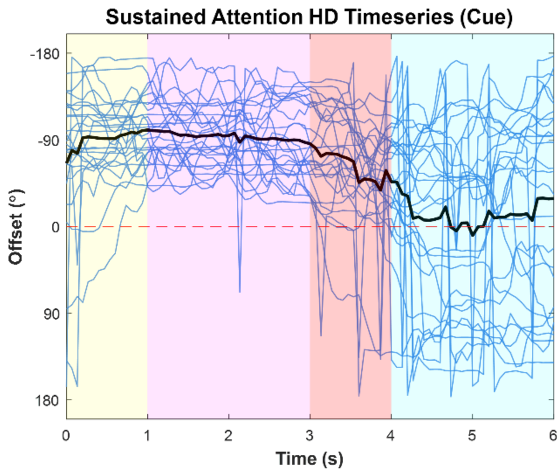
To examine the attention states of the observer mice under the tone trigger during observational learning, we also analyzed the dynamics of the head directions of the observer mice during the observational learning session in cohort 2. The top part of Figure 22 presents the timeseries of head directions of the observer mice during a total of 6 seconds of the observational trials. This includes the last 1 second of the demo mice approaching the reward zone, the 2 seconds that the demo mice stayed in the reward zone, the 1 second that the high-frequency tone was being played, and the first 2 seconds that the demo mice returned to the cage after completing the spatial task, across a total of 112 observational trials in cohort 2.

Although plotting all the timeseries curves on a single figure would make it look cluttered (Figure 22, top), upon closer inspection of each head direction timeseries, we discovered that when the demo mice were reaching the reward zone, the observer mice would significantly face towards the maze, triggered by the cue. Based on the timeseries of head direction, we identified two stereotypical types of attention during the observational learning process: 'transferable attention' and 'sustained attention'.

'Transferable attention' is characterized by the observer mouse facing towards the food dispensing outlet located behind the cage before the tone is played, and turning its head direction towards the maze in front of the cage when the tone is heard. 'Sustained attention' is characterized by the observer mouse consistently facing towards the maze in front of the cage before and after the tone is played. Among the total of 112 observational trials in cohort 2, we identified 37 trials of 'transferable attention' (Figure 22, middle left) and 26 trials of 'sustained attention' (Figure 22, middle right). The remaining 49 trials with less stable head direction timeseries were categorized as 'inattentiveness'. The bottom part of Figure 22 shows the distribution of these three types of attention in cohort 2. From these results, we can conclude that using a high-frequency tone as a cue is more effective in drawing the observer mouse's attention to the demo mouse's performance during observational learning, compared to dispensing a food reward simultaneously to the observer mouse.







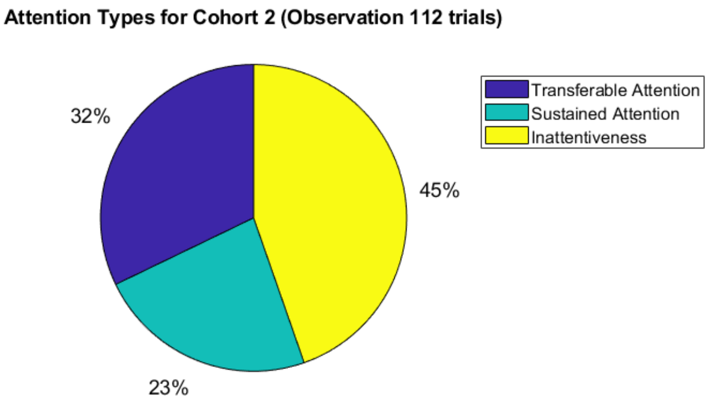


Figure 22: (top) The timeseries of head direction of the observer mice during a total of 6 seconds, including 4 phases, across a total of 112 observational trials in cohort 2. Blue lines represent the head direction timeseries for observer mice in individual trial, while the black line represents the average of head direction angles across all trials. (middle left) Head direction curves of 36 trials which show transferable attention towards the tone. (middle right) Head direction curves of 26 trials which show sustained attention during the process of the tone being played. (bottom) The distribution of the three types of attention in cohort 2.

To more clearly compare the differences in head direction angles across different attention types, we compared the 39 trials of 'sustained attention to reward' from cohort 1, the 37 trials of 'transferable attention' from cohort 2, and the 26 trials of 'sustained attention' also from cohort 2. Each trial provided a total of 91 data points, each representing a head direction angle from one frame of the 6-second timeseries. As shown in Figure 23, the head direction angles of the 'sustained attention to reward' type of attention in cohort 1 were mainly concentrated around 90°, indicating that the mice were primarily facing the food dispenser located behind the cage. In contrast, the head direction angles of the 'transferable attention' type of attention in cohort 2 shifted from around 90° to around -90° when the tone was played, indicating that the mice turned their attention from the food dispenser located behind the cage to the maze located in front of the cage upon hearing the tone. The head direction angles of the 'sustained attention' type of attention in cohort 2 were mainly concentrated around -90°, indicating that the mice were primarily facing the maze located in front of the cage both before and after the tone was played.

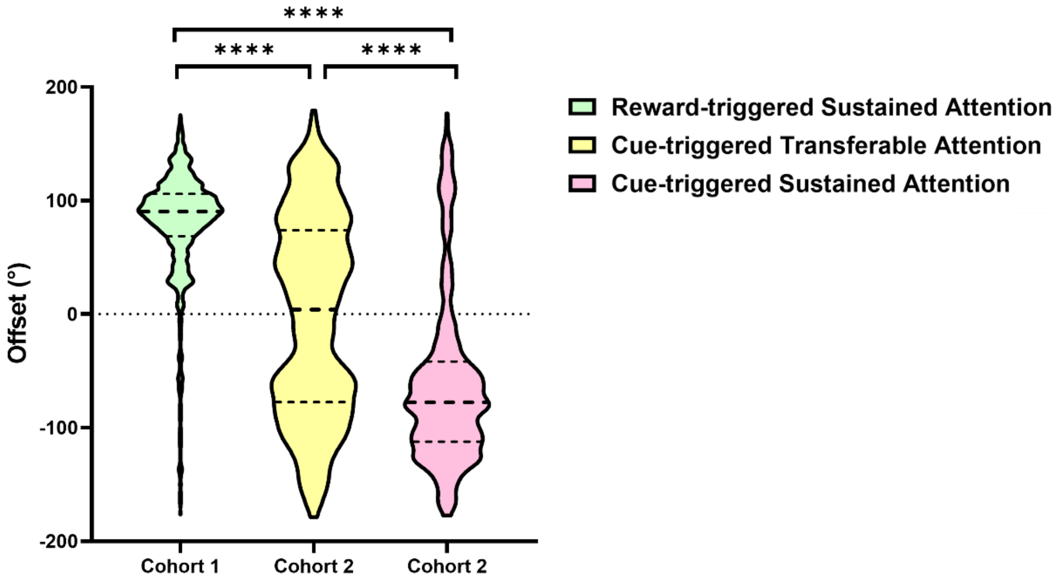


Figure 23: A violin plot depicting the distribution statistics of head direction angles during observational trials for three types of attention (medians with quartiles indicated). 'Reward-triggered sustained attention' (green) includes 3549 data points from 39 trials in cohort 1, 'cue-triggered transferable attention' (yellow) includes 3367 data points from 37 trials in cohort 2, and 'cue-triggered sustained attention' (pink) includes 2366 data points from 26 trials in cohort 2.

3.2.4. Performance enhancement through observation

In light of the results from cohort 1, which showed that the performance variance of the observer mice in the testing trials was large when they lacked prior experience with the spatial task, we made some modifications in cohort 2. Not only did we train the observer mice to undergo the single mouse training session to remember a reward position different from that of the demo mice, but we also placed a food reward in the reward zone of the demo mouse on the maze during the testing trials in the first two days of the observational learning session.

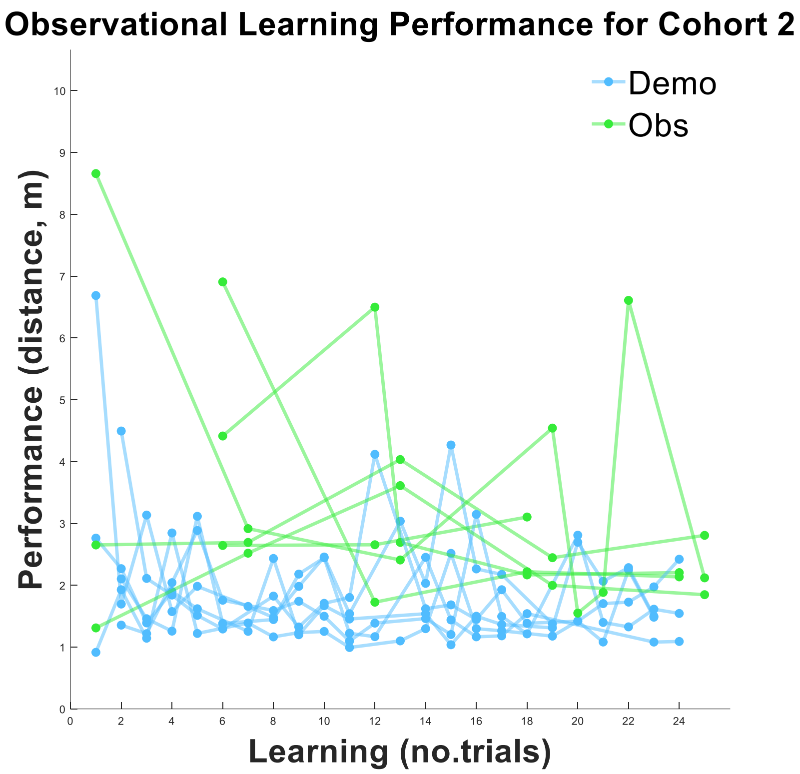


Figure 24: Learning performance curves of observational learning in cohort 2, including the learning performance curves for 3 pairs of demo mice (blue) and observer mice (green) across 2 consecutive days during the observational learning sessions.

Similar to the observational learning performance in cohort 1 (Figure 20), we also plotted the performance of the demo mice in the observational trials and the observer mice in the testing trials in cohort 2 for the first two days of training (Figure 24). As can be seen from Figure 24, with prior knowledge and the trigger of a food reward on the maze, the observer mice quickly remapped their memorized goal position to the new reward position of the demo mice (green lines in Figure 24) in the testing trials. From this, we realized that placing a food reward on the maze for the observer mice greatly reduced the difficulty of completing the testing trial, thereby making the effect of observational learning less salient. Therefore, we decided that in subsequent observational learning sessions, we would only train the observer mice to undergo the same single mouse training session to familiarize them with the spatial task, and we would not place a food reward for the observer on the maze during the testing trials of the observational learning session.

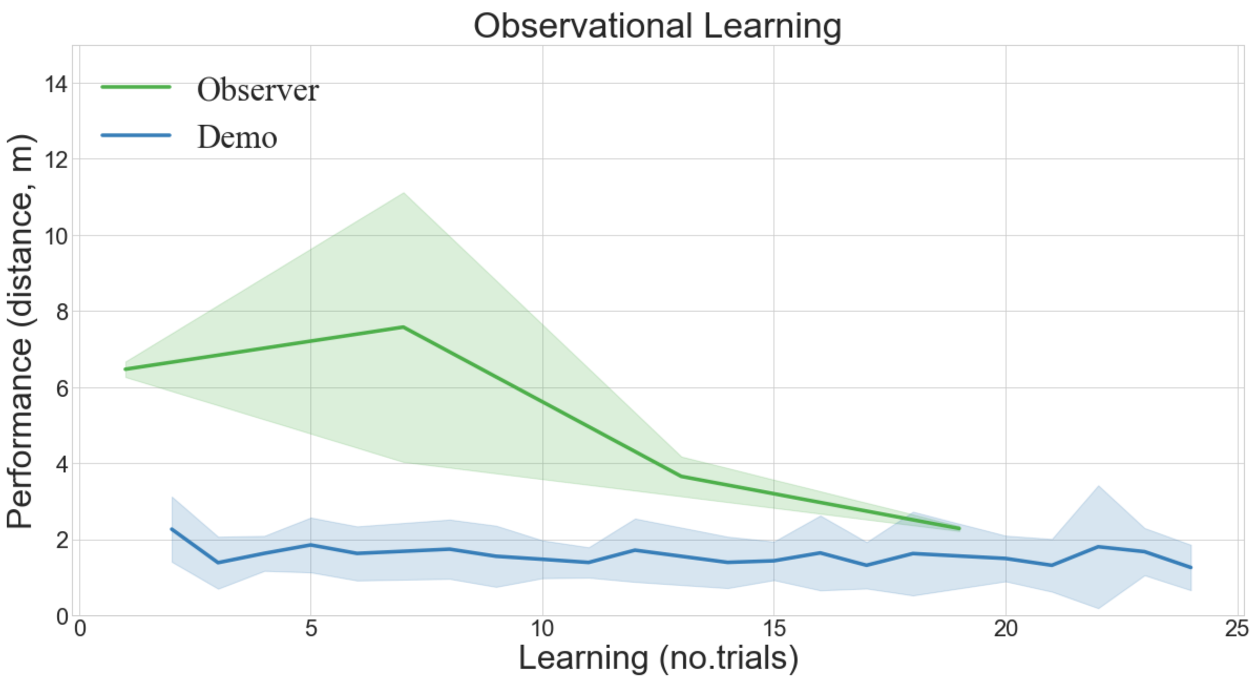


Figure 25: Learning performance curves of observational learning from both cohort 1 and cohort 2. The figure includes the learning performance curves for all 6 pairs of demo mice (blue) and observer mice (green) from cohort 1 and all 2 pairs from cohort 2. Shaded areas represent standard deviation.

Thus, on the third day of the observational learning session in cohort 2, we ceased placing a food reward for the observer on the maze during the testing trials. Both observer mice showed improved performance in the testing trials as the number of observational trials progressed, indicating a smooth remapping to the new reward location. In Figure 25, we plotted these 2 performance curves from cohort 2, absence of food reward for the observer on the maze during the testing trials. As depicted by the green line in Figure 25, a consistent trend of improvement in the observer mice's performance in the testing trials was evident as the observational trials continued. However, we remained uncertain if the observer mice's incremental improvement was attributable to observational learning or if it stemmed from their individual learning experiences through exploration. Consequently, we decided to establish a control group from cohort 3 to specifically investigate the role of observation in mice's spatial learning.

3.2.5. Observational learning task with a control group

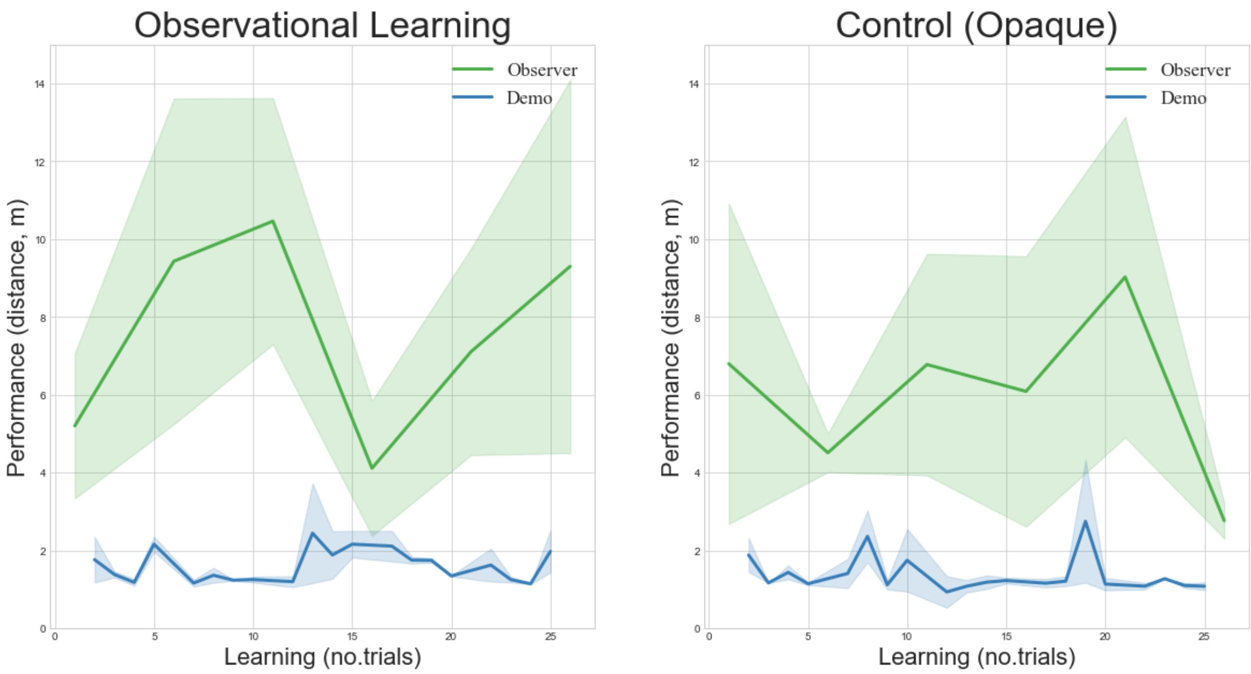


Figure 26: Learning performance curves of observational learning for both the observational group (left) and control group (right) in cohort 3. Each plot includes the learning performance curves for 3 pairs of demo mice (blue) and observer mice (green). Shaded areas indicate standard deviation.

During the three consecutive days of the observational learning session in cohort 3, we established a control group where the observer's cage was enclosed with opaque cardboard. This setup ensured that the observer in the control group was deprived of visual observational input during the observational learning trials. However, in both the observational and control groups, the auditory tone was still played when the demo mouse successfully completed the task. Each day's training consisted of one pair of demo mice and observer mice for both the observational and control groups. From the results presented in Figure 26, with the observational group on the left and the control group on the right, each containing results from three pairs of performance curves for demo mice and observer mice, there was no significant difference in the performance of the observer mice between the experimental and control groups. Given the considerable variance in mouse behaviour in the absence of food on the maze, this outcome might be attributed to the limited number of testing trials.

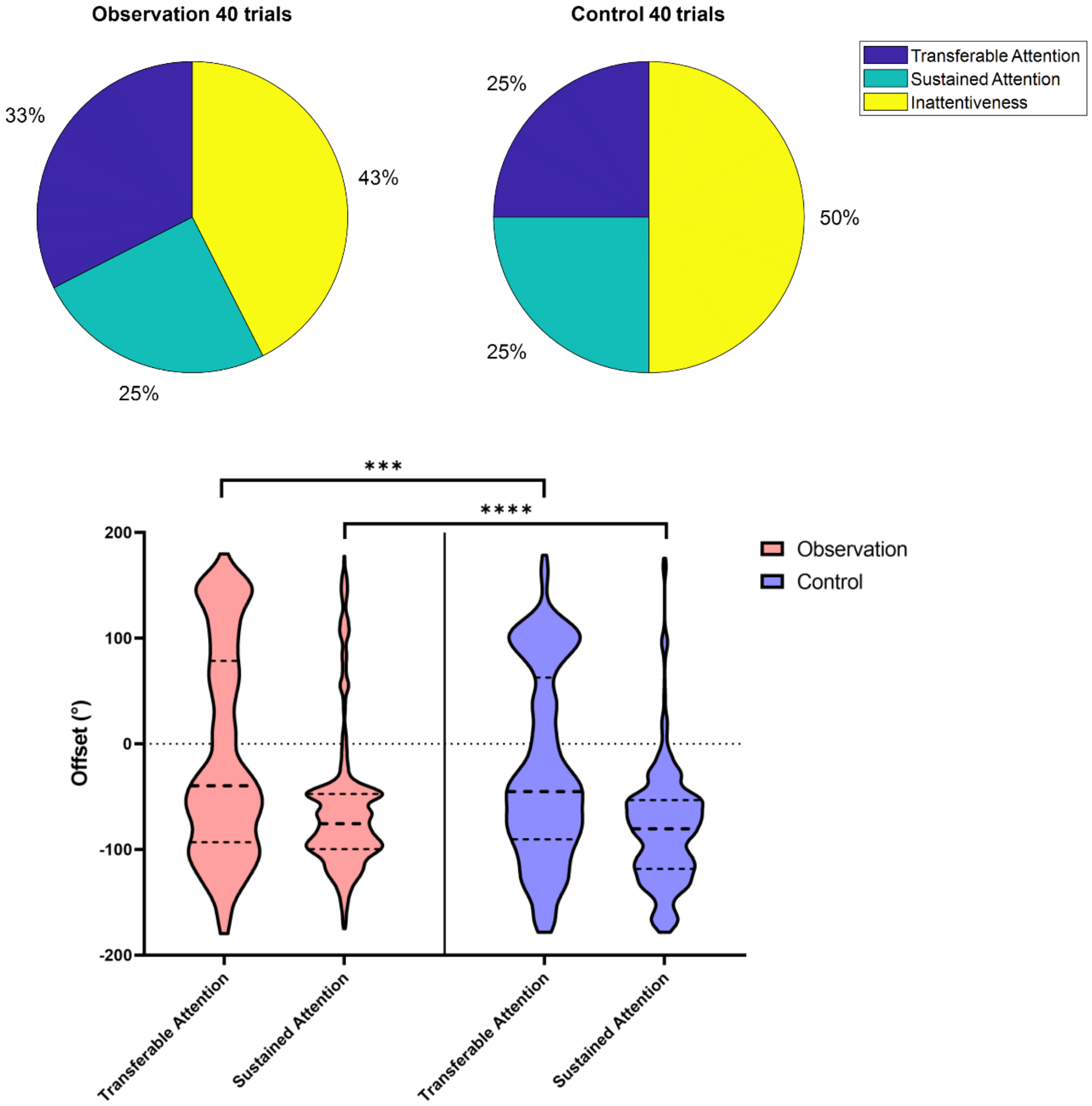


Figure 27: (top) Distribution of the three attention types for the observational group (left) and control group (right) in cohort 3. (bottom) Violin plot illustrating the distribution statistics of head direction angles during observational trials for the two attention types in both the observational group (red) and control group (blue) in cohort 3, with medians and quartiles indicated.

Following a similar analysis procedure as before, we conducted a head direction analysis to compare the differences in the attention states of the observer mice between the observational group and the control group in cohort 3. The top part of Figure 27 illustrates the distribution proportions of the three previously defined attention types, as observed over two days of the observational learning session in cohort 3 for both groups. Notably, there isn't a significant difference between the observational and control groups. However, the observational group does show a slightly higher proportion (33 %) of 'transferable attention'. The bottom part of Figure 27 compares the differences in head direction angles among the different attention types in both the observational and control groups. This encompasses 13 trials of 'transferable attention' from the observational group, 10 trials of 'sustained attention' from the observational group, 10 trials of 'transferable attention' from the control group, and 10 trials of 'sustained attention' from the control group, all observed over two days of the observational learning session in cohort 3. Each trial provided a total of 91 data points, each representing a head direction angle from one frame of the 6-second timeseries. Although statistical t-tests suggest the presence of significant differences, the violin plot distribution in the bottom part of Figure 27 does not clearly highlight any discernible differences in head direction angles between the observational and control groups. These results indicate that the observer mice in the control group also paid attention to the maze by turning their head direction upon hearing the tone. This finding suggested that mice in both the observational group and the control group formed a conditioned memory associating the auditory tone with the spatial reward during the previous single mice training session.

4. Discussion

Our main goal was to study the reward-directed spatial observational learning in mice. To this end, we developed a training paradigm centered on reward-directed spatial learning using a cheeseboard maze. Initially, we trained demo mice to master this spatial task, examining their learning performance under various conditions. Subsequently, we proceed to the observer mice, focusing on their attention states during the observation of the demo mice and their performance changing during observation.

Our hypotheses drew inspiration from previous research, which revealed that animals can gain both positive and negative survival experiences by observing their peers (1). Existing literature suggested that during observational learning, observer rats, might undergo a neural replay of the demo rat's path in their hippocampal place cells, a neural activity that subsequently guides their performance when they undertake the same task (29). Moreover, the neural activity within the hippocampus of mice is found to guide both spatial and social learning (13). Therefore, our first hypothesis was that a single mouse, through training, could accurately encode and retrieve long-term spatial memories. Our second hypothesis was that observer mice could enhance their performance by observing demo mice navigate the spatial task.

The results of our single mouse training on the goal-oriented spatial task largely support our first hypothesis. The learning performance quantified by trajectory distance and navigation latency to the reward position showed that a single mouse could form stable spatial memories and performance after undergoing several days of behavioural shaping even in the absence of immediate rewards. This suggests that, over multiple days of repetitive spatial task training, the mouse might has encoded spatial locations by establishing stable place fields of its hippocampal place cells, and formed a conditioned association between completing a specific spatial task and obtaining a subsequent delayed reward.

However, in the observational learning session, not all observer mice displayed a progressively improving performance during the testing trials inserted within observational trials, which goes against our second hypothesis. Several factors could account for these unexpected results. The limited number of testing trials, the short duration of the observational learning session, or the limitations in the long-distance visual range of mice could all play a role. Additionally, our assumption that providing a simultaneous reward to the observer mouse would help it associate the demo mouse's spatial performance with the reward was disproved. Instead, the observer mouse seemed to merely await its reward, paying rare attention to the demo mouse's behaviour. However, not all results from the mouse observational learning session were negative. We discerned that if observer mice were trained in the single mouse training session to familiarize them with the task structure and were conditioned to associate an auditory cue with the reward, they would significantly pay attention to the spatial maze task upon hearing the cue during observational learning.

In summation, while the single mouse training session demonstrated the mouse's ability to form long-term stable spatial memories, the observational learning session did not yield a pronounced difference between the observational and control groups. The only clear indicator of the mouse's attention was its head direction, especially when it heard an auditory cue previously associated with a reward. Future studies might benefit from employing miniscope imaging to study the place cell activity in observer mice as they attentively observe the demo mouse. Perhaps there's a correlation between their place field and the reward location of the demo mouse they subsequently remap to. Additionally, extending the observation period by introducing a pure observation session before testing trials, increasing the number of testing trials to reduce variance, or modifying the maze's design could yield more definitive results. Furthermore, to ensure that observer mice actively observe the demo mouse's performance, we could consider introducing operational conditions, adjusting the maze's design, or even physically positioning the observer mice to ensure they face the maze.

5. References

1. Olsson, Andreas, Ewelina Knapska, and Björn Lindström. "The neural and computational systems of social learning." *Nature Reviews Neuroscience* 21.4 (2020): 197-212.

2. Carcea, Ioana, and Robert C. Froemke. "Biological mechanisms for observational learning." *Current opinion in neurobiology* 54 (2019): 178-185.

3. Burgos-Robles, Anthony, et al. "Conserved features of anterior cingulate networks support observational learning across species." *Neuroscience & Biobehavioral Reviews* 107 (2019): 215-228.

4. Carlier, Pascal, and Marc Jamon. "Observational learning in C57BL/6j mice." *Behavioural brain research* 174.1 (2006): 125-131.

5. Carcea, Ioana, et al. "Oxytocin neurons enable social transmission of maternal behaviour." *Nature* 596.7873 (2021): 553-557.

6. Omer, David B., et al. "Social place-cells in the bat hippocampus." *Science* 359.6372 (2018): 218-224.

7. Jercog, Pablo E., et al. "Heading direction with respect to a reference point modulates place-cell activity." *Nature communications* 10.1 (2019): 2333.

8. Dupret, David, et al. "The reorganization and reactivation of hippocampal maps predict spatial memory performance." *Nature neuroscience* 13.8 (2010): 995-1002.

9. Priestley, James B., et al. "Signatures of rapid plasticity in hippocampal CA1 representations during novel experiences." *Neuron* 110.12 (2022): 1978-1992.

10. Zhang, Huanqiu, et al. "Hippocampal spatial representations exhibit a hyperbolic geometry that expands with experience." *Nature Neuroscience* 26.1 (2023): 131-139.

11. Oliva, Azahara, et al. "Hippocampal CA2 sharp-wave ripples reactivate and promote social memory." *Nature* 587.7833 (2020): 264-269.

12. Dragoi, George, and Susumu Tonegawa. "Preplay of future place cell sequences by hippocampal cellular assemblies." *Nature* 469.7330 (2011): 397-401.

13. Murugan, Malavika, et al. "Combined social and spatial coding in a descending projection from the prefrontal cortex." *Cell* 171.7 (2017): 1663-1677.

14. Jurado-Parras, M. Teresa, Agnès Gruart, and José M. Delgado-García. "Observational learning in mice can be prevented by medial prefrontal cortex stimulation and enhanced by nucleus accumbens stimulation." *Learning & memory* 19.3 (2012): 99-106.

15. Kim, Seong-Wook, et al. "Hemispherically lateralized rhythmic oscillations in the cingulate-amygdala circuit drive affective empathy in mice." *Neuron* (2022).

16. Phillips, Hannah L., et al. "Dorsomedial prefrontal hypoexcitability underlies lost empathy in frontotemporal dementia." *Neuron* (2023).

17. Allsop, Stephen A., et al. "Corticoamygdala transfer of socially derived information gates observational learning." *Cell* 173.6 (2018): 1329-1342.

18. Jeon, Daejong, et al. "Observational fear learning involves affective pain system and Cav1. 2 Ca2+ channels in ACC." *Nature neuroscience* 13.4 (2010): 482-488.

19. Keum, Sehoon, and Hee-Sup Shin. "Neural basis of observational fear learning: a potential model of affective empathy." *Neuron* 104.1 (2019): 78-86.

20. Jiang, Wan-Chen, Shengjin Xu, and Joshua T. Dudman. "Hippocampal representations of foraging trajectories depend upon spatial context." *Nature neuroscience* 25.12 (2022): 1693-1705.

21. Sosa, Marielena, and Lisa M. Giocomo. "Navigating for reward." *Nature Reviews Neuroscience* 22.8 (2021): 472-487.

22. Turi, Gergely Farkas, et al. "Vasoactive intestinal polypeptide-expressing interneurons in the hippocampus support goal-oriented spatial learning." *Neuron* 101.6 (2019): 1150-1165.

23. Gauthier, Jeffrey L., and David W. Tank. "A dedicated population for reward coding in the hippocampus." *Neuron* 99.1 (2018): 179-193.

24. Jarzebowski, Przemyslaw, et al. "Different encoding of reward location in dorsal and intermediate hippocampus." *Current Biology* 32.4 (2022): 834-841.

25. Deshmukh, Sachin S., and James J. Knierim. "Hippocampus." *Wiley Interdisciplinary Reviews: Cognitive Science* 3.2 (2012): 231-251.

26. Ormond, Jake, and John O’Keefe. "Hippocampal place cells have goal-oriented vector fields during navigation." *Nature* 607.7920 (2022): 741-746.

27. Danjo, Teruko, Taro Toyoizumi, and Shigeyoshi Fujisawa. "Spatial representations of self and other in the hippocampus." *Science* 359.6372 (2018): 213-218.

28. Omer, David B., Liora Las, and Nachum Ulanovsky. "Contextual and pure time coding for self and other in the hippocampus." *Nature neuroscience* 26.2 (2023): 285-294.

29. Mou, Xiang, et al. "Observational learning promotes hippocampal remote awake replay toward future reward locations." *Neuron* 110.5 (2022): 891-902.

30. Mathis, Alexander, et al. "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning." *Nature neuroscience* 21.9 (2018): 1281-1289.

31. Willmore, Lindsay, et al. "Behavioural and dopaminergic signatures of resilience." *Nature* (2022): 1-9.

32. Zhang, Huanqiu, et al. "Hippocampal spatial representations exhibit a hyperbolic geometry that expands with experience." *Nature Neuroscience* 26.1 (2023): 131-139.

33. Sit, Kevin K., and Michael J. Goard. "Coregistration of heading to visual cues in retrosplenial cortex." *Nature communications* 14.1 (2023): 1992.

34. Asumbisa, Kadjita, Adrien Peyrache, and Stuart Trenholm. "Flexible cue anchoring strategies enable stable head direction coding in both sighted and blind animals." *Nature Communications* 13.1 (2022): 5483.

35. Finkelstein, Arseny, et al. "Optimal dynamic coding by mixed-dimensionality neurons in the head-direction system of bats." *Nature communications* 9.1 (2018): 3590.

36. Vouros, Avgoustinos, et al. "A generalised framework for detailed classification of swimming paths inside the Morris Water Maze." *Scientific reports* 8.1 (2018): 15089.

Appendix A. MATLAB code for task control

```
% Set the path for output data
Directory = 'D:\yihui\CBM_data\';   % Main directory\
date = 'day7_Nov_23_2022';          % Date\
mouse_name = 'mouse_LLR';           % Mouse name\
type = 'demo';                      % Mouse type (demo or obs)\

% Initialize the door and food dispenser controlled by Arduino
delete(instrfindall);               % Delete all existing serial port objects
s = serial('COM4');                 % Connect to serial port (COM4)
set(s,'BaudRate',9600);             % Set baud rate (communication speed in bits per second)
set(s,'Timeout',30);                % Set time out (allowed time in seconds to complete read and write)
set(s,'InputBufferSize',8388608);   % Set input buffer size

fopen(s);                           % Open connection to Arduino board
if (exist('board1','var'))          % Stop the running program in Arduino
   board1.stop;pause(0);
end

% Open video input (maze)
imaqreset                                                % Disconnect and delete all image acquisition objects
imaqmex('feature','-limitPhysicalMemoryUsage',false);    % Set unlimited physical memory usage

vid_maze = videoinput('winvideo', 1, 'RGB24_744x480'); % Create video input for the camera above maze
set(vid_maze,'Timeout',35);                            % Set time out (allowed time to wait for image)
vid_maze.FramesPerTrigger = Inf;                       % Set frames per trigger to infinity
vid_maze.ReturnedColorspace = 'grayscale';             % Set color space to gray scale
src_maze = getselectedsource(vid_maze);                % Get current video source object
src_maze.FrameRate = '15.0000';                        % Set frame rate to 15 fps
src_maze.ExposureMode = 'manual';                      % Set exposure mode to manual
src_maze.Exposure = -7;                                % Set exposure value to -7
src_maze.GainMode = 'manual';                          % Set gain mode to manual
src_maze.Gain = 16;                                    % Set gain value to 16

% Get and save the background image
start(vid_maze);                                       % Start maze video collection
pause(3)                                               % Wait 3 secs for video configuration
background = getsnapshot(vid_maze);                    % Get one background image of the maze
stop(vid_maze);                                        % Stop maze video collection

Data_Folder = [Directory date '\' mouse_name '_' type '\']; % Set data folder name
if ~exist(Data_Folder,'dir')                                % Create data folder if it doesn't exist
   mkdir(Data_Folder)
end
save([Data_Folder 'background.mat'],'background');          % Save the background image of the maze
```

```
% Set the area coordinates of inner boundary for left cage
figure(1);                              % Create a figure window to show background image
disp('Please draw the inner boundary for left cage, and double click to save the coordinates    ');
bw_in_left=roipoly(background);         % Interactively create a region for inner boundary of left cage
[r_in_left,c_in_left]=find(bw_in_left==1);  % Get the row and column coordinates of all points in inner region

left_in_x1 = min(c_in_left);            % Get the horizontal coordinate of the first vertex of left region
left_in_y1 = min(r_in_left(c_in_left == min(c_in_left)));
                                        % Get the vertical coordinate of the first vertex of left region
left_in_x2 = min(c_in_left(r_in_left == min(r_in_left)));
                                        % Get the horizontal coordinate of the second vertex of left region
left_in_y2 = min(r_in_left);            % Get the vertical coordinate of the second vertex of left region

left_in_x3 = max(c_in_left);            % Get the horizontal coordinate of the third vertex of left region
left_in_y3 = max(r_in_left(c_in_left == max(c_in_left)));
                                        % Get the vertical coordinate of the third vertex of left region
left_in_x4 = max(c_in_left(r_in_left == max(r_in_left)));
                                        % Get the horizontal coordinate of the fourth vertex of left region
left_in_y4 = max(r_in_left);            % Get the vertical coordinate of the fourth vertex of left region

coordinates_in_left = [left_in_x1 left_in_x2 left_in_x3 left_in_x4; ...
   left_in_y1 left_in_y2 left_in_y3 left_in_y4]';
                                        % Get the coordinates of all four vertices of left inner region

% Set the area coordinates of outer boundary for left cage
figure(1);                              % open the figure window showing background image
disp('Please draw the outer boundary for left cage, and double click to save the coordinates    ');
bw_out_left=roipoly(background);        % Interactively create a region for outer boundary of left cage
[r_out_left,c_out_left]=find(bw_out_left==1); % Get row and column coordinates of all points in outer region

left_out_x1 = min(c_out_left);          % Get the horizontal coordinate of the first vertex of left region
left_out_y1 = min(r_out_left(c_out_left == min(c_out_left)));
                                        % Get the vertical coordinate of the first vertex of left region
left_out_x2 = min(c_out_left(r_out_left == min(r_out_left)));
                                        % Get the horizontal coordinate of the second vertex of left region
left_out_y2 = min(r_out_left);          % Get the vertical coordinate of the second vertex of left region

left_out_x3 = max(c_out_left);          % Get the horizontal coordinate of the third vertex of left region
left_out_y3 = max(r_out_left(c_out_left == max(c_out_left)));
                                        % Get the vertical coordinate of the third vertex of left region
left_out_x4 = max(c_out_left(r_out_left == max(r_out_left)));
                                        % Get the horizontal coordinate of the fourth vertex of left region
left_out_y4 = max(r_out_left);          % Get the vertical coordinate of the fourth vertex of left region

coordinates_out_left = [left_out_x1 left_out_x2 left_out_x3 left_out_x4; ...
   left_out_y1 left_out_y2 left_out_y3 left_out_y4]';
                                        % Get the coordinates of all four vertices of left outer region
```

```
% Open video input (cages)
vid_cage = videoinput('tisimaq_r2013_64', 4, 'RGB24 (640x480) [Binning 2x]');
                                                  % Create video input for the camera above cage
set(vid_cage,'Timeout',35);                       % Set time out (allowed time to wait for image)
vid_cage.FramesPerTrigger = Inf;                  % Set frames per trigger to infinity
vid_cage.ReturnedColorspace = 'grayscale';        % Set color space to gray scale
src_cage = getselectedsource(vid_cage);           % Get current video source object
src_cage.ExposureAuto = 'Off';                    % Set exposure mode to manual
src_cage.Exposure = 0.0666;                       % Set exposure value to 0.0666
src_cage.GainAuto = 'Off';                        % Set gain mode to manual
src_cage.Gain = 0;                                % Set gain value to 0

FrameRate = 15;                                   % Set frame rate parameter to 15 fps

% Set the center coordinates for rewards
figure(1);                        % Open the figure window showing background image
imshow(background);
reward_center = ginput(1);        % Interactively select a point for reward zone center

reward_on_maze = 1;               % Whether food are placed on maze: true(1) / false(0)
reward_radius = 45;               % Radius of reward zone: 45 pixels (10 cm)
duration_range = 3;               % Search mouse position in previous 3 secs from current time point
reward_duration_range = duration_range*FrameRate;        % Number of frames in previous 3 secs
duration_threshold = 2;           % Mouse needs to stay in reward zone for 2 secs
reward_duration_threshold = duration_threshold*FrameRate; % Number of frames for the 2 secs threshold

% Set the tracking parameters
trial_length = 600;               % Set longest duration for one trial to 600 secs
exploration_threshold = 180;      % End the trial after 180 secs of exploration without reaching reward zone
initialization_threshold = 60;    % End the trial after 60 secs without mouse's leaving the cage

thresh = 0.4;                     % Threshold for converting image to binary image
cage_in_thresh = 20000;           % Threshold for testing whether mouse enters the inner boundary of cage
cage_out_thresh = 36000;          % Threshold for testing whether mouse leaves the outer boundary of cage

% Initiate the trial
trial_num = '1';                  % Change the number for every trial

trial = ['trial' trial_num];      % Convert the trial number to a string

Data_Folder = [Directory date '\' mouse_name '_' type '\' trial '\'];    % Set data folder name for every trial

if exist(Data_Folder,'dir')       % Ask user if they want to overwrite the data folder if it exists
  promptMessage = sprintf('This directory already exists:\n%s\nDo you want to overwrite it?', Data_Folder);
  titleBarCaption = 'Overwrite?';
  buttonText = questdlg(promptMessage, titleBarCaption, 'Yes', 'No', 'Yes');
  if strcmpi(buttonText, 'No')
```

```
        error('trial number already exists') % Display error message if user does not want to overwrite
    else
        close;                          % Close the question dialog box if user want to overwrite
    end
else
    mkdir(Data_Folder)                  % Create data folder if it doesn't exist
end

% Set the Tone Waveform
A = 2;                                  % Amplitude
f_0 = 4000;                             % Frequency of sound (4k Hz)
fs = 40000;                             % Sampling frequency (40k Hz)
N = 40000;                              % Signal sampling points number, Playback duration (1 secs)
y = A*sin(2*pi*f_0*(0:N-1)/fs);         % Single frequency sine signal
Speaker_volume = 33;                    % Speaker volume

% Open video output
writerObj=VideoWriter([Data_Folder 'processed_behavior.avi']); % Create object to write processed video
writerObj.FrameRate = FrameRate;                               % Set frame rate for output video to 15 fps
open(writerObj);                                               % Open output video object

% Save the tracking parameters.
centroids = zeros(1,2);               % Centroids to temporarily save coordinates of mouse in each frame
centers = zeros(trial_length*FrameRate,2); % Centers to save coordinates of mouse for all frames

i = 1;                                % Parameter i records the currently analyzed frame number
T = zeros(1,trial_length*FrameRate); % T record the exact time when each frame was captured

step_timepoint = zeros(1,6);          % Record the exact time when each operation step was performed

Step_2 = 1;                           % Set the initial state of step 2 to True
Step_4 = 1;                           % Set the initial state of step 4 to True
Step_5 = 1;                           % Set the initial state of step 5 to True

reward_duration = 0;       % Record the duration in number of frames for mouse staying in the reward zone
reward_timepoint = zeros(1,trial_length*FrameRate);
                           % Record the frame number when mouse stays in the reward zone

% Start the trial
start(vid_maze);                        % Start maze video collection
t_vid_maze_Start = tic;                 % Record the exact time when maze video collection was started

start(vid_cage);                        % Start cage video collection
t_vid_cage_Start = tic;                 % Record the exact time when cage video collection was started

tStart = tic;                           % Start timer to measure elapsed time
```

```
fprintf(s,'left_door_open/');                    % Step 1: Open the left cage door to release the Demo mouse
step_timepoint(1)= toc(tStart);                  % Record the exact time for step 1
disp('Step 1: Open the left cage door to release the Demo mouse    ');

% Start capturing video frames in real time
while toc(tStart) < trial_length                 % Capture video frames before exceeding longest trial duration

   ROI_frame = getsnapshot(vid_maze);            % Acquisite one frame from maze video
   T(i)= toc(tStart);                            % Record the exact time when every frame was captured
   i = i + 1;                                    % Parameter i records the currently analyzed frame number

   ROI_Im = imabsdiff(ROI_frame,background);
                                 % Subtract the background image from the current frame to extract mouse shape

   cage_in_left = ROI_Im.*uint8(bw_in_left);     % Extract the left inner cage image
   cage_out_left = ROI_Im.*uint8(bw_out_left);   % Extract the left outer cage image

   figure(1);                                    % Open figure window 1
   imshow(ROI_frame);                            % Show the current frame
   hold on                                       % Hold on the plotted figure

   % Plot rectangular regions for two cages and circular area for reward
   patch('XData',[left_in_x1 left_in_x2 left_in_x3    left_in_x4],'YData',[left_in_y1 left_in_y2 left_in_y3
left_in_y4],'EdgeColor','yellow','FaceColor','none','LineWidth',1);
                                                 % Plot a yellow rectangular region for the left inner cage

   patch('XData',[left_out_x1 left_out_x2 left_out_x3 left_out_x4],'YData',[left_out_y1 left_out_y2 left_out_y3
left_out_y4],'EdgeColor','green','FaceColor','none','LineWidth',1);
                                                 % Plot a green rectangular region for the left outer cage

   plot(reward_center(:,1),reward_center(:,2), 'b.'); % Plot a blue dot in the reward center
   viscircles(reward_center,reward_radius,'Color','blue','LineWidth',0.4,'LineStyle','--');
                                                 % Plot a blue circle as the reward zone

   % Detection of mouse in cage
   if sum(cage_in_left,'all') > cage_in_thresh          % mouse is detected in the cage
      patch('XData',[left_in_x1 left_in_x2 left_in_x3 left_in_x4],'YData',[left_in_y1 left_in_y2 left_in_y3
left_in_y4],'EdgeColor','yellow','FaceColor','yellow','FaceAlpha',0.5,'LineWidth',1);
                                                 % Fill the inner rectangle with semitransparent yellow
  end

   if sum(cage_out_left,'all') < cage_out_thresh        % mouse is detected out the cage
      patch('XData',[left_out_x1 left_out_x2 left_out_x3 left_out_x4],'YData',[left_out_y1 left_out_y2
left_out_y3 left_out_y4],'EdgeColor','green','FaceColor','green','FaceAlpha',0.3,'LineWidth',1);
                                                 % Fill the outer rectangle with semitransparent green
   end
```

```matlab
% Find the mouse center
I = im2bw(ROI_Im,thresh);           % Convert the subtracted image to binary image, based on threshold
k = regionprops('table',I,'Area');   % Measure the area(number of pixels) of binary image regions
idx = find(max([k.Area]));           % Get the index of the largest region
cc = bwconncomp(I);                  % Find and count all connected components in the binary image
g = ismember(labelmatrix(cc), idx);% Get the binary image of only the largest region
m = regionprops('table',g,'Area','Centroid','MajorAxisLength','MinorAxisLength');
                                     % Measure the size properties of the largest region
if(size(m,1)==1)                     % If only one region survives
   centroids = cat(1, m.Centroid);  % Get the coordinates of mouse
   centers(i,:) = centroids;         % Save the coordinates of mouse for this frame in centers
   plot(centroids(:,1),centroids(:,2), 'r.')                    % Plot a red dot in the mouse center
   diameters = mean([m.MajorAxisLength m.MinorAxisLength],2); % Calculate mouse region's diameter
   radii = diameters/2;                                         % Calculate the radius of the mouse region
   viscircles(centroids,radii,'LineWidth',0.1);                 % Plot a red circle as the mouse position
end
hold off                             % Hold off the plotted frame

frame2 = getframe;                   % Get the plotted frame
writeVideo(writerObj,frame2);        % Write the plotted frame into the processed video

% Control the door and food dispenser
if Step_2                                % The step 2 operation has not been performed
   if sum(cage_out_left,'all') < cage_out_thresh % Mouse is detected out the cage
      fprintf(s,'left_door_close/');          % Step 2: Close the left cage door when Demo mouse has left
      step_timepoint(2)= toc(tStart);         % Record the exact time for step 2
      disp('Step 2: Close the left cage door, Demo mouse is leaving the left cage    ');
      Step_2 = 0;                             % Set the state of step 2 to False

   elseif toc(tStart) > initialization_threshold     % After waiting for maximum 1 mins
      fprintf(s,'left_door_close/');          % Close the cage door to end the trial
      step_timepoint(2)= toc(tStart);         % Record the exact time for step 2
      disp('Step 2: After 1 min, close the cage door to end the trial    ');
      Step_2 = 0;                             % Set the state of step 2 to False
      break                                   % Break from the loop of capturing video frames
   end

% (Step 3: Provide food for the observer mice, when the Demo mouse found the reward)

elseif Step_4                            % The step 4 operation has not been performed
   if pdist([reward_center ; centroids],'euclidean') < reward_radius    % The mouse is in the reward zone
      reward_timepoint(i) = 1;           % Mark the frame number when mouse stays in the reward zone
      reward_duration = length(find(reward_timepoint(max([i-reward_duration_range,1]):i)));
                       % Calculate the duration in number of frames for mouse staying in the reward zone
      if reward_duration > reward_duration_threshold      % Mouse has stayed in reward zone for 2 secs
         fprintf(s,'left_food/');                         % Provide food for the Demo mouse;
         step_timepoint(4)= toc(tStart);                  % Record the exact time for step 4
```

```
            disp('Step 4: Demo mouse is in the reward zone for 2 secs, provide food for the demo mouse  ');

            % play tone for pellet dispense (1 secs)
            sound(y,fs);                        % Sound playback through sound card
            Step_4 = 0;                         % Set the state of step 4 to False

            fprintf(s,'left_door_open/');       % Step 4: Open the cage door to let the mouse back to cage
            disp('Open the left cage door to let the mouse back to cage    ');
        end

    elseif toc(tStart) > exploration_threshold % After exploring for maximum 3 mins
        fprintf(s,'left_door_open/');           % Step 4: Open the cage door to let the mouse back to cage
        step_timepoint(4)= toc(tStart);         % Record the exact time for step 4
        disp('Step 4: After 3 mins, open the cage door to let the mouse back to cage    ');
        Step_4 = 0;                             % Set the state of step 4 to False
    end

  elseif Step_5                          % The step 5 operation has not been performed
    if sum(cage_in_left,'all') > cage_in_thresh % Mouse is detected in the cage
        fprintf(s,'left_door_close/');       % Step 5: Close the left cage door when Demo mouse is in cage
        step_timepoint(5)= toc(tStart);      % Record the exact time for step 5
        disp('Step 5: Close the left cage door, Demo mouse is in the left cage    ');
        Step_5 = 0;                          % Set the state of step 5 to False
        pause(3)                             % Wait for 3 secs
        fprintf(s,'left_door_close/');       % Close the cage door again in case the operation went wrong;
        pause(7)                             % Wait for 7 secs
        break                                % Break from the loop of capturing video frames
    end
  end
end

tEnd = toc(tStart);                          % Get the exact ending time
step_timepoint(6)= tEnd;                     % Record the exact ending time
disp(['Elapsed time: ',num2str(tEnd),' seconds.']); % Display the exact ending time

close(writerObj);                            % Close output video object

stop(vid_maze);                              % Stop maze video collection
t_vid_maze_End = toc(t_vid_maze_Start); % Record the exact time when maze video was ended

stop(vid_cage);                              % Stop cage video collection
t_vid_cage_End = toc(t_vid_cage_Start);  % Record the exact time when cage video was ended

close(figure(1));                            % Close the figure window

% save maze video
data = getdata(vid_maze, vid_maze.FramesAvailable);   % Get all collected frames of maze video
```

```
numFrames = size(data, 4);                  % Get number of frames

diskLogger = VideoWriter([Data_Folder 'shaping_to_maze.avi'], 'Uncompressed AVI');
                                            % Create object to write the uncompressed maze video
diskLogger.FrameRate = numFrames./t_vid_maze_End; % Set the frame rate
open(diskLogger);                           % Open output video object
for ii = 1:numFrames
   writeVideo(diskLogger, data(:,:,:,ii));   % Write each frame into the maze video
end
close(diskLogger);                          % Close output video object

% save cage video
data = getdata(vid_cage, vid_cage.FramesAvailable);   % Get all collected frames of cage video
numFrames = size(data, 4);                  % Get number of frames

diskLogger = VideoWriter([Data_Folder 'shaping_to_cage.avi'], 'Uncompressed AVI');
                                            % Create object to write the uncompressed cage video
diskLogger.FrameRate = numFrames./t_vid_cage_End; % Set the frame rate
open(diskLogger);                           % Open output video object
for ii = 1:numFrames
   writeVideo(diskLogger, data(:,:,:,ii));   % Write each frame into the cage video
end
close(diskLogger);                          % Close output video object

% save data files
save([Data_Folder 'centers.mat'],'centers'); % Save coordinates of mouse for all frames
save([Data_Folder 'step_timepoint.mat'],'step_timepoint');    % Save the exact time for each operation step
save([Data_Folder 'parameters.mat'],'reward_center','reward_radius','duration_threshold','reward_on_maze','re-
ward_timepoint','exploration_threshold','coordinates_in_left','coordinates_out_left','T','Speaker_volume');
                                                          % Save other parameters
```

Appendix B. Arduino codes for hardware control

B.1. Master Arduino board controlling food dispensers

```
#include<Wire.h>              // This library is used for I2C communication

int left_Address = 2;         // left_Address here is the address of the left slave board
int right_Address = 3;        // right_Address here is the address of the right slave board

int x;                        // x value sets the rotation direction of stepper motor

int right_Pin = 12;           // Set the digital pin 12 to control right food dispenser
int left_Pin = 13;            // Set the digital pin 13 to control left food dispenser

int Delay = 1000;             // Set 1000 millisecond delay after each pellet is dispensed
char terminator = '/';        // Set terminator for each input command
String mode;                  // mode string saves the command mode

// Setup code are put here to run once:
void setup() {
  Serial.begin(9600);         // Begin serial communication (baud rate 9600) with MATLAB
  Wire.begin();               // Activate I2C link

  pinMode(right_Pin, OUTPUT);   // sets the digital pin 12 as right output
  pinMode(left_Pin, OUTPUT);    // sets the digital pin 13 as left output

  digitalWrite(right_Pin, LOW); // Initiate the right output for the dispense operation
  digitalWrite(left_Pin, LOW);  // Initiate the left output for the dispense operation

  // Print instructions in the serial monitor
  Serial.println("<Arduino is ready>");
  Serial.println("-----------------------------------");
  Serial.println("Command List:");
  Serial.println("-----------------------------------");
  Serial.println("COMMAND: Left/Right food dispenser drop one pellet");
  Serial.println("SYNTAX:  left_food/");
  Serial.println("SYNTAX:  Right_food/");
  Serial.println("-----------------------------------");
  Serial.println("COMMAND: Left/Right door open/close");
  Serial.println("SYNTAX:  left_door_open/");
  Serial.println("SYNTAX:  left_door_close/");
  Serial.println("SYNTAX:  Right_door_open/");
  Serial.println("SYNTAX:  Right_door_close/");
  Serial.println("-----------------------------------");
}
```

```
// Main code are put here to run repeatedly:
void loop() {
  while (Serial.available() == 0) {          // Wait for user input
  }
  mode = Serial.readStringUntil(terminator);  // Read the input command from MATLAB to set mode

  if (mode == "left_food")                    // The mode is dispensing food into left cage
  {
    digitalWrite(left_Pin, HIGH);             // Set the digital left pin to HIGH
    delay(Delay);                             // Wait for a second
    digitalWrite(left_Pin, LOW);              // Release and wait for the next dispense operation
  }
  if (mode == "Right_food")                   // The mode is dispensing food into right cage
  {
    digitalWrite(right_Pin, HIGH);            // Set the digital right pin to HIGH
    delay(Delay);                             // Wait for a second
    digitalWrite(right_Pin, LOW);             // Release and wait for the next dispense operation
  }
  if (mode == "left_door_open")               // The mode is opening the left cage door
  {
    x = 2;                                    // The value 2 of x means to rotate the stepper motor counterclockwise
    Wire.beginTransmission(left_Address);     // Begin communication with the left slave board
    Wire.write(x);                            // Transfers the x value to the left slave board
    Wire.endTransmission();                   // End communication with the left slave board
  }
  if (mode == "left_door_close")              // The mode is closing the left cage door
  {
    x = 1;                                    // The value 1 of x means to rotate the stepper motor clockwise
    Wire.beginTransmission(left_Address);     // Begin communication with the left slave board
    Wire.write(x);                            // Transfers the x value to the left slave board
    Wire.endTransmission();                   // End communication with the left slave board
  }
  if (mode == "Right_door_open")              // The mode is opening the right cage door
  {
    x = 1;                                    // The value 1 of x means to rotate the stepper motor clockwise
    Wire.beginTransmission(right_Address);    // Begin communication with the right slave board
    Wire.write(x);                            // Transfers the x value to the right slave board
    Wire.endTransmission();                   // End communication with the right slave board
  }
  if (mode == "Right_door_close")             // The mode is closing the right cage door
  {
    x = 2;                                    // The value 2 of x means to rotate the stepper motor counterclockwise
    Wire.beginTransmission(right_Address);    // Begin communication with the right slave board
    Wire.write(x);                            // Transfers the x value to the right slave board
    Wire.endTransmission();                   // End communication with the right slave board
  }
}
```

B.2. Slave uStepper board controlling animal cage doors

```
#include <uStepperS.h>                  // This library is used for controlling stepper motor
#include<Wire.h>                        // This library is used for I2C communication

#define NODE_ADDRESS 2                  // The unique address for each I2C slave board

uStepperS stepper;                      // Create an object of the uStepper S
float up_angle = 1040.0;                // Amount of degrees to move up
float down_angle = -1040.0;             // Amount of degrees to move down

int x;                                  // x value sets the rotation direction of stepper motor

boolean down_flag = true;               // Boolean value record the down status of cage door
boolean up_flag = true;                 // Boolean value record the up status of cage door


// Setup code are put here to run once:
void setup() {
  stepper.setup();                      // Initialisation of the uStepper S
  stepper.setMaxAcceleration(8000);     // Set an acceleration of 2000 fullsteps/s^2
  stepper.setMaxVelocity(2000);         // Set max velocity of 500 fullsteps/s

  Serial.begin(9600);                   // Begin serial communication (baud rate 9600) with MATLAB
  Wire.begin(NODE_ADDRESS);             // Activate I2C network
  Wire.onReceive(receiveEvent);         // Set the slave node to receive value from master board
}

void receiveEvent(int bytes) {          // Define a founction to receive value from master board
  while (Wire.available()) {
    x = Wire.read();                    // Receive value from master board
  }
}

// Main code are put here to run repeatedly:
void loop() {
  if (x == 1) {                         // The value 1 of x means to move down the left cage door
    if (down_flag) {                    // If the cage door is still closed
      if (!stepper.getMotorState())     // If motor is at standstill
      {
        stepper.moveAngle(down_angle);  // Start to move down
        down_flag = false;              // Set the down status of cage door to false
        up_flag = true;                 // Set the up status of cage door to true
        delay(1000);                    // Wait for a second
      }
    }
  }
  if (x == 2) {                         // The value 2 of x means to move up the left cage door
    if (up_flag) {                      // If the cage door is still open
      if (!stepper.getMotorState())     // If motor is at standstill
      {
        stepper.moveAngle(up_angle);    // Start to move up
        up_flag = false;                // Set the up status of cage door to false
        down_flag = true;               // Set the down status of cage door to true
        delay(1000);                    // Wait for a second
      }
    }
  }
}
```