Mouse Academy: high-throughput automated training and trial-by-trial behavioral analysis during learning

Mu Qiao[1], Tony Zhang[1], Cristina Segalin[2], Sarah Sam[1], Pietro Perona[2,3] & Markus Meister[1,3*]

[1]Division of Biology and Biological Engineering, California Institute of Technology, Pasadena, CA 91125, USA

[2]Division of Engineering and Applied Sciences, California Institute of Technology, Pasadena, CA 91125, USA

[3]Tianqiao and Chrissy Chen Institute for Neuroscience, California Institute of Technology, Pasadena, CA 91125, USA

*Correspondence: meister@caltech.edu

ABSTRACT

Progress in understanding how individual animals learn will require high-throughput standardized methods for behavioral training but also advances in the analysis of the resulting behavioral data. In the course of training with multiple trials, an animal may change its behavior abruptly, and capturing such events calls for a trial-by-trial analysis of the animal's strategy. To address this challenge, we developed an integrated platform for automated animal training and analysis of behavioral data. A low-cost and space-efficient apparatus serves to train entire cohorts of mice on a decision-making task under identical conditions. A generalized linear model (GLM) analyzes each animal's performance at single-trial resolution. This model infers the momentary decision-making strategy and can predict the animal's choice on each trial with an accuracy of ~80%. We also introduce automated software to assess the animal's detailed trajectories and body poses within the apparatus. Unsupervised analysis of these features revealed unusual trajectories that represent hesitation in the response. This integrated hardware/software platform promises to accelerate the understanding of animal learning.

INTRODUCTION

Learning – the change of neural representation and behavior that results from past experience and the consequences of actions – is important for animals to survive and forms a central topic in neuroscience[1]. Different individuals may apply different strategies to the learning process, reflecting their individual personalities. Indeed, substantial differences in sensory biases, locomotion, motivation, and cognitive competence have been observed in populations of fruit flies[2,3], rodents and primates[4-6]. Thus, it is critical to investigate learning at the individual level.

Rodents, especially the mouse, have become popular experimental animals in studying associative learning and decision-making, because of the wide availability of transgenic resources[7-10]. They can learn to perform complex decision-making tasks that probe cognitive components such as working memory and selective attention[11-13]. However, differences in learning strategies across individuals have rarely been addressed, partly owing to the limitations of data gathering and analysis.

Studying differences among individuals requires training and collecting data from multiple animals in a standardized and high-throughput fashion. The training procedures are often time-consuming, requiring several days to many weeks[8,9], depending on the task. Although there have been advances in training automation, existing systems either require an experimenter to move animals from the home cage to the training apparatus[14-16], or training animals within their own cages[17-19]. The former introduces additional sources of variability[20,21], and the latter precludes tasks that require a large training arena. Following data acquisition, the analysis of behavior aims at understanding the learning process. Present approaches tend to focus on the averaged performance over many trials[22]. However, changes in behavior may happen at a single trial, and thus the modeling of behavior should similarly offer a time resolution of single trials to assess each animal's individual approach to learning.

To address these challenges, we present Mouse Academy, an integrated platform for automated training of group-housed mice and analysis of behavioral changes in learning a decision-making task. We designed hardware that makes use of implanted radio frequency identification (RFID) chips to identify each mouse, and guides the animal into a behavior training box. Synchronized video recordings and decision-making sequences are acquired during animal learning. To analyze the decision-making sequences, we developed an iterative generalized linear model (GLM). This model makes a prediction of the animal's choice in each trial and gets updated based on the animal's actual choice. This iterative GLM model achieves a prediction accuracy of ~80%, and also reveals the decision-making strategy of the animal and how it changes over time. To analyze the animal's behavior during the task in greater detail, we developed automated software that tracks the animal in video recordings and extracts its location and body pose using deep convolutional neural networks (CNNs). These features allowed us to perform an unsupervised analysis of each animal's behavior, and discover individual traits of behavioral learning that were not apparent from the simple choice sequences.

RESULTS

The Mouse Academy platform consists of three components (**Fig. 1**): an automated RFID sorting and animal training system, an iterative GLM to analyze decision-making sequences, and behavior assessment software that extracts animal trajectories from video data.

Automated RFID sorting supports individual training programs

We designed the equipment in the following manner (**Fig. 1a**): RFID-tagged mice are grouped in a common home cage where food and bedding is supplied. The home cage connects to a behavior training box through a gated tunnel. The gates are controlled by a home-made RFID animal sorting system[23]: three RFID antennas are placed along the tunnel, with one near the home cage, one near the training box and one between the two; the motorized gates are placed between the RFID sensors, separating the tunnel into three compartments. An Arduino microcontroller integrates information from the RFID readers to open and shut the gates, allowing only one animal at a time to pass through the tunnel (**Supplementary Figs. 1a, 1c, 1d** and **Supplementary Videos 1-4**). The behavior box is outfitted with three ports, each of which contains a photo-transistor to detect snout entry, a solenoid valve to deliver water reward, and a light emitting diode (LED) to present visual cues. To maintain a controlled environment, the training box is isolated from the outside by a light- and sound-proof chamber (**Supplementary Fig. 1b**).

Once a mouse enters the training box, a protocol is set up to train the mouse to perform a certain task. In the experiments reported here, the animal must nose-poke the center port to initialize a trial and then hold the position for a short period. Visual or auditory stimuli are delivered, and based on these stimuli, the animal must choose to poke one of the side ports. If the correct response is chosen, the animal gets water reward from a lick tube in the response port, otherwise a timeout punishment is applied. This training process is controlled by Bpod, an Arduino microcontroller that interfaces with the three ports. Data from the response ports as well as video recordings from an overhead camera are acquired simultaneously as the animal is trained.

The entire apparatus is orchestrated by a master program that coordinates the RFID sorting device, the Bpod system, synchronized video recording, data management and logging (**Supplementary Fig. 1e**). The program monitors the amount of water each animal consumes per day and regulates the time each animal can spend in the training box per session. In addition, the software updates the training protocol for each animal based on its performance, for example switching to a harder task once a simpler one has been mastered (**Supplementary Fig. 2**). This lets each animal learn at its own pace.

The apparatus can be assembled at a materials cost of $1500-2500, with the cheaper option using a Raspberry Pi computer as the controller (**Supplementary Fig. 1f** and **Supplementary Table 1**). Compared with designs in which each animal is automatically trained in its own home cage[15,17], the system saves considerable space. Because housing and training are independent modules, the same system can be used for diverse training environments.

We tested the automated RFID sorting and animal training system by training group-house mice to learn a variety of decision tasks, following similar procedures as reported previously[11,12] (**Supplementary Fig. 2** and Online methods). The training period lasted 28 days, with up to five mice in the common home cage. Each animal occupied the training box for 3-4 hours per day (13-15% of the 24 hours) throughout the entire training period (**Figs. 2a**, **2b** and **Supplementary Fig. 3**). For a sample cohort of four animals trained in sessions of 90 trials each, we found that the behavior box was occupied most of the time, with brief empty intervals of <10 min (**Figs. 2c, 2d** and **2e**). Each animal was trained for over 900 trials (10 sessions), and consumed more than 1.9 mL of water per day (**Fig. 2f**). Interestingly there was no circadian pattern to the animals' training activity, even though the setup was illuminated on a daily light cycle (12 h on / 12 h off) (**Fig. 2g**). As observed previously, it appears that animals working for a goal can avoid circadian modulation of the locomotor pattern[24,25].

A generalized linear model accurately predicts decision-making during training

In a decision-making task, an animal is asked to associate distinct stimuli with distinct responses. Although this is the ultimate goal, during learning, it is often observed that the animal begins by basing its decisions on unrelated input variables and gradually switches to using the stimulus variables that actually predict reward. We define a policy as a mapping of these variables to the animal's decisions. A fundamental goal in the study of learning is to infer what policy the animal follows at any given time and to determine how the policy evolves with experience.

We applied a generalized linear model (GLM) to map factors relevant to the animal's decision-making to its choices through logistic regression. A common way to build such a GLM is by fitting data of an entire session[16,26]. However, this loses resolution in single trials within the session. During learning, a change of policy can happen at each trial. Thus, we developed the model to make trial-by-trial choice predictions based on various factors the animal might plausibly use. The model works in an iterative two-step process (**Fig. 1b**). In the prediction step, the model makes a prediction for the next decision based on the input factors. Once the outcome of the animal's decision is observed, an error term between the model's prediction and the observation is computed. This error, after weighting by a reward factor and a temporal discount factor, is fed back to the loss function. In the update step, the model is updated by minimizing the regularized loss function. This iteration happens after every trial. The temporal discount factor accounts for the possibility that the most recent trials impact the current decision more than remote trials. The reward factor accounts for the fact that water rewards and timeout punishments may have effects of different magnitude on the updates of the animal's policy.

We illustrate the utility of this model by fitting results from an easy visual task, in which one of the two choice ports lights up to indicate the location of the reward, and the optimal policy is to simply poke the port with the light (**Supplementary Fig. 2a, 2a'** and **2a''**). All the mice eventually reached a >83% performance level, comparable to what mice achieve in similar tasks[19,27]. The GLM makes a prediction for the outcome of each trial based on a weighted combination of several input variables: the current visual stimulus, a constant bias term, and three terms representing the history of previous trials (**Fig. 3a**). These inputs from a previous trial include the port choice, whether that choice was rewarded, and a term indicating the multiplicative interaction between the choice and reward (Choice x Reward). This term supports a strategy called win-stay-lose-switch (WSLS), which chooses the same port if it was rewarded previously and the opposite one if not. Since a GLM cannot multiply two inputs, we provided this interaction term explicitly. Each of the above terms has a weight coefficient that can be positive or negative. For instance, a positive weight for the visual stimulus supports turning towards the light, and a negative weight away from the light.

To determine the extent of trial history that affects the animal's behavior, we fitted the model to the response data including history-dependent terms up to three previous trials. We found that only the immediately preceding trial had an appreciable effect on the prediction accuracy, and thus restricted further analysis to those inputs (**Fig. 3b**). The model also has three hyperparameters (the temporal discount factor α, the reward factor r, and the regularization factor λ), and we optimized them for each animal by grid search. We found that each animal had a different set of hyperparameters, reflecting differences in the learning process across individuals (**Fig. 3c**). Among the four sample mice, Animal 2 had the lowest temporal discount factor, suggesting that it weighed recent trials more heavily and updated the policy more quickly. Indeed, this is the animal that learned the fastest among the four (**Fig. 3d**).

Predictions from the iterative GLM matched ~80% of the animals' actual choices (**Fig. 3f**), and the predicted accuracy of each animal captured the actual fluctuations of its learning curve (**Fig. 3e** and **Supplementary Fig. 4**). We compared the performance of the GLM with two other modeling approaches (Online methods). The first model was fit to the animal's average performance in the task; its trial-by-trial match of the animal's actual choices was only ~59%

(**Fig. 3g**). The second model was a logistic regression fitted to data in a sliding window of N trials. This sliding window model performed worse than the iterative GLM when the window size was small ($N = 20$ and 30 trials, **Fig. 3h**); for larger windows the performance was comparable. Overall, the iterative model is advantageous because it makes predictions online as every trial occurs and adapts dynamically to the growing data set.

Individual learning policies can be inferred from iterative GLM fitting

The iterative GLM serves to infer what policy the animal follows in making decisions. The linear weight of each input term reflects its relative importance for the decision. By following this weight vector across trials one obtains a policy matrix that documents how the animal's policy changes during learning (**Figs. 1b** and **4c**). To test that the model can correctly capture a time-varying policy, we simulated decision-making data from a ground truth policy that changed at a certain frequency, including a certain level of noise in the behavioral output (**Fig. 4a**). Over a wide range of policy change frequencies and noise levels, the GLM was able to capture the ground truth policy (**Figs. 4a** and **4b**). In addition, different values of policy change frequency and noise levels led to different sets of hyperparameters fitted from the model, showing that the GLM can adapt to individuals with diverse learning characteristics (**Supplementary Figs. 5a-e**).

We then recovered the policy matrix of each animal from the GLM fits. All four animals started with the non-optimal policy of WSLS. Subsequently each animal followed its own learning process (**Fig. 4c**): Animal 2 had a clear bias towards the right port at the beginning but it rapidly found the optimal policy of following the light. The other three animals were slower learners. Animal 3 and Animal 4 followed similar processes to converge to the optimal policy. Animal 1 was distinct from the others. At the early stages, it had a strong bias towards the left port and it made decisions based on whether the previous choice was rewarded.

We further validated the transition between policies during learning by analyzing the first and last sessions of each animal and counting how many choices could be explained by each policy (**Fig. 4d**). Indeed, we found a clear switch from the (non-optimal) WSLS policy to the (optimal) stimulus-based policy (**Fig. 4e** and **Supplementary Fig. 5f**). The animals might have been biased towards the WSLS strategy by a shaping method we used during training, which offered the animal a repeat of the same stimulus every time it made a mistake (Online methods). To test whether these correlations in the trial sequence influenced the final policy we performed two additional analyses. First, we only included trials following a correct trial, and performed logistic regression on these trials for each session. This analysis showed that at least on these trials, all the animals based their decisions on the light stimulus by the end of learning (**Supplementary Fig. 6a**). Second, we compared the error rate on trials following an incorrect choice with that following a correct one. We found no significant difference between the two error rates during the last session (**Supplementary Figs. 6b** and **6c**), suggesting that the animals treated these two types of trials identically.

Automated movement tracking reveals fine structure of behavioral responses

Thus far the report has focused on the animal's responses only as sensed by the nose pokes into response ports. The GLM fits of those responses already revealed differences in policy across individuals. To gain further insight into these individual preferences, it is essential to track each animal's behavior along the way from stimuli to responses[10]. We thus developed software that uses deep learning to automatically, quantitatively and accurately assess each animal's behavior during decision-making (**Figs. 1c** and **5a**).

To track the animal location and body coordinates, we recorded videos of the animal from above, and analyzed them with a sequence of two deep convolutional neural networks (CNNs) that were pre-trained on annotated pose data (**Fig. 5a** and **Supplementary Videos 5-7**). The first CNN was based on the multi-scale convolutional (MSC) Multibox detector[28] (**Supplementary Fig. 7a**). For each frame of the video it computed a crop frame around the body of the mouse. The second CNN was a Stacked Hourglass Network[29] that used the cropped video frame to locate seven body landmarks: the nose, the ears, the neck, the body sides and the tail of the animal (**Supplementary Figs. 7a** and **7b**). These landmarks allowed precise identification of the animal's position and body pose (**Supplementary Fig. 7c**), from which we further extracted two features: the body centroid (average position of the seven landmarks) and the orientation (angle of the line connecting the centroid and the nose).

To illustrate use of these behavioral trajectories, we focus on the period of the visual choice task where the animal reports its decision: from the time it leaves the center port to when it pokes one of the side ports. The trials fall into four groups based on location of the stimulus and the response. As expected, the trajectories of position and orientation clearly distinguish left from right choices (**Figs. 5b** and **5d**). Interestingly, the trajectories also reveal whether the decision was correct: On incorrect decision, the trajectories reversed direction after ~0.5 s, because the animal quickly turned back to the center after finding no reward in the chosen port (**Figs. 5c, 5e** and **Supplementary Video 5**). A linear kernel support vector machine (SVM), trained to predict the category of each trial from a 1 s trajectory, was able to correctly distinguish correct and incorrect choices with an accuracy of over 90% (**Supplementary Fig. 8**). In addition, many of the trajectories were highly asymmetric and again revealed differences across individuals. For instance, Animal 2 and Animal 4 started from a location close to the right port, Animal 1 closer to the left port (**Fig. 5c**). This asymmetry correlates with the bias revealed by the iterative GLM: each animal prefers to select the port closer to its body location.

Unsupervised behavioral analysis reveals moments of hesitation

Whereas the supervised learning discussed above relies on prior classification of stimuli and responses, an unsupervised analysis has the potential to discover unexpected structures in the animal's behavior[30]. We thus performed an unsupervised classification of the behavioral trajectories.

After subjecting all the trajectories of a given animal to principal component analysis (PCA) we projected the data onto the top three components, which explained over 95% of the variance (**Figs. 6a, 6b** and **Supplementary Fig. 9a**). Importantly, without any labels from trial types, these three PCs captured meaningful features that differentiated the animal's responses. The first PC separated movements to the left from those to the right (**Figs. 6a** and **6b**). The third PC captured the turning-back behavior after an incorrect choice (**Fig. 6b** and **Supplementary Fig. 9a**). The second PC captured different baseline positions (**Fig. 6b**). Each animal has its own preference for a baseline position somewhere off the midline of the chamber (**Supplementary Fig. 9b**).

We also projected the trajectories into 2 dimensions using a non-linear embedding method, t-distributed stochastic neighbor embedding[31,32] (t-SNE). Unlike PCA, this graph prioritizes the preservation of local structures within the data instead of the global structure[32]. In the t-SNE space the trajectories formed clear clusters (**Fig. 6c**). Most of the clusters are dominated by one of the decision categories (**Fig. 6c** and **Supplementary Fig. 9c**). Interestingly, we found clusters in Animals 2, 3, and 4, in which the centroid trajectories were flat, unlike the trajectories of the four decision categories (**Fig. 6d**), suggesting that animals hesitated in these trials and made decisions only after a delay. Indeed, in trials flagged by these clusters, the animals had longer reaction times (**Fig. 6e**). Furthermore, such hesitating responses were more common following an incorrect trial (**Fig. 6f**); they may reflect a behavioral adjustment to prevent further mistakes[33].

DISCUSSION

Despite the fact that rodents can be trained to perform interesting decision-making tasks[7-10], the learning progress of individual animals has rarely been addressed. Doing so requires training and observing many animals in parallel under identical conditions, and the ability to analyze the decision policy of each animal on a trial-by-trial basis. To meet these demands, we developed Mouse Academy, an integrated platform for automated training and behavior analysis of individual animals.

We demonstrate here that Mouse Academy can train group-housed mice in an automated and highly efficient manner while simultaneously acquiring decision-making sequences and video recordings. Automated animal training has been of great interest in recent years and efforts have focused on two directions. In one design, multiple animals are trained in parallel within stacks of training boxes. This requires a technician to transfer animals from their home cages to the behavior boxes[14-16]. Such animal handling has been reported to introduce additional variability[20,21], and even the mere presence of an experimenter can influence behavioral outcomes[34]. Thus, eliminating the requirement for human intervention, as in Mouse Academy, likely reduces experimental variation. In another design, a training setup is incorporated within the animals' home cage[17-19]. By contrast, Mouse Academy separates the functions of housing and training, and that modular design allows easy adaptation to a different purpose. For instance, one can replace the 3-port discrimination box with a maze to study spatial navigation learning[35,36], or with an apparatus for training under voluntary head-fixation[37]. In each case, a single training apparatus can serve many mice, potentially from multiple home cages.

To understand how an animal's decision-making policies change in the course of learning, we developed a trial-by-trial iterative GLM. The evolution of the model is similar to online machine learning[38] in which the data are streamed in sequentially, rather than in batch mode. The linear nature of the model supports a straightforward definition of the animal's decision policy, namely as the vector of weights associated with different input variables. In addition, the simple linear structure allows rapid execution of the algorithm, which favors its use in real-time closed-loop behavior experiments. The model also allows several parametric adjustments. One specifies how much the recent trials are weighted over more distant ones in shaping the animal's policy. Another rates the relative influence of reward versus punishments. Fitting these parameters to each animal already revealed differences in learning style. This model can have a broader use beyond mouse decision-making, for instance to track the progress of human learners from their answers to a series of quizzes[39].

Finally we presented software for automated assessment of behaviors based on video recordings within Mouse Academy. Largely extending existing methods, the software uses deep convolutional neural networks for animal tracking and pose estimation. Because the animal's movements are unconstrained, we performed the tracking in two stages: the first finds the animal within the video frame and the second locates the body landmarks. Compared to a single-shot procedure, this split approach requires fewer learning examples and less computation in the second stage[38]. The resulting behavioral trajectories can reveal intricate aspects of the animal's decision process that are hidden from a mere record of the binary choices. The large data volume again calls for automated analysis, and both supervised machine learning methods[30,40,41] and unsupervised classification[30-32,42] have been employed for this purpose. Unsupervised analysis is not constrained by class labels, and can identify hidden structure in the data in an unbiased manner. In the present case, we discovered a motif wherein the animal hesitates on certain trials before taking action.

Mouse Academy can be combined with chronic wireless recording[43,44], to allow synchronized data acquisition of neural responses. Researchers can seek correlations between neural activity and the policy matrix or even the behavioral trajectories. This will open the door to a mechanistic understanding of how neural representations and dynamics change in the course of animal learning.

ONLINE METHODS

Animals

Subjects were C57BL/6J male mice aged 8-12 weeks. All experiments were conducted in accordance with protocols approved by the Institutional Animal Care and Use Committee of the California Institute of Technology.

Hardware setup

The hardware setup comprises a behavioral training box, an engineered home cage, and a radio frequency identification (RFID) sorting system, which allows animals to move between the home cage and the training box. These components are coordinated by customized software.

The design file for the behavior box was modified from that of Sanworks LLC (https://github.com/sanworks/Bpod-CAD) using Solid-Works computer-aided design software and the customized behavioral training box was manufactured in the lab. The behavior box is controlled by a Bpod state machine (r0.8, Sanworks LLC). To monitor the animal's behavior, an IR webcam (Ailipu Technology or OpenMV Camera M7) is installed above the behavior box. The behavior box and the webcam are placed within a light- and sound-proof chamber. The chamber is made of particle board with walls covered by acoustic foam. A tunnel made of red plastic tubes connects the behavior box to a home cage (**Supplementary Fig. 1b**).

For the RFID access control system, an Arduino Mega 2560 microcontroller is connected with three RFID readers (ID-12LA, Sparkfun) with custom antenna coils spaced along the access tunnel. The microcontroller controls two generic servo motors fitted with plastic gates to grant individual access to the training box (**Supplementary Fig. 1a**).

The microcontroller identifies each animal by its implanted RFID chip and permits only one animal to go through the tunnel connecting the home cage and the behavioral training box (**Supplementary Fig. 1c)**. It also communicates the animal's identity to a master program running on a PC or Raspberry Pi (Matlab or Python). The master program coordinates the following programs: Bpod (https://github.com/sanworks/Bpod), synchronized video recording, data management and logging. A repository containing the design files, the firmware code for the microcontroller, and the software can be found in https://github.com/muqiao0626/Mouse_Academy.

Behavior training

The training procedures of mice to perform a selective attention task are similar to those previously reported[11,12]. Mice were water restricted for seven days before training, and habituated in the automated training system to collect reward freely for several sessions. Then the mice were trained in sessions, each of which was made of 90 trials, to collect water rewards by performing two alternative forced choice tasks. Briefly, the animal had to nose-poke one of two choice ports based on the presented stimuli. If the decision was correct, 10% sucrose-sweetened water (3 μL) was delivered to the animal. For incorrect responses, the animal was punished with a five-second timeout. Following an incorrect response, the animal was presented with the identical trial again; this simple shaping procedure helps counter-act biases in the behavior.

Over 28 days of training the animals learned increasingly complex tasks, from visual discrimination to a two-modality cued attention switching task[11,12]. The training progressed through six stages (**Supplementary Fig. 2**):

1. A simple visual task: In this task, the animal initiates a trial by poking the center port and holding the position for 100 ms. Then either the left or right side port light up briefly until the animal moves away from the center port. The animal must then poke one of the two side ports within the decision period of 10 s. Choice of the port flagged by the light leads to a water reward, and choice of the other port leads to a time-out period during which no trials can be initiated. Data presented in the main text are from this stage of training only.
2. A simple auditory task: As Stage 1, except that the stimulus was white noise sound either the left or the right side to flag the reward port.
3. A cued single-modality (visual or auditory) switching task: Blocks of 15 trials consisting of single-modality (visual or auditory) stimulus presentation. Each block was like stages 1 or 2, except that the trial type was indicated by a 7 kHz (visual) or 18 kHz (auditory) pure tone.
4. A cued single- and double-modality switching task: Like stage 3, but distracting trials were introduced in which both visual and auditory stimuli were present, but only one of the modalities was relevant to the decision. The relevant modality was again indicated by the pure tone cues. In repeating blocks, four types of trials were presented: a. five visual-only trials; b. ten 'attend to vision' trials with auditory distractors; c. five auditory-only trials; d. ten 'attend to audition' trials with visual distractors. During the training, the time that the animal had to hold in the center port was gradually increased to 0.5 s, and the duration of the stimuli was gradually shortened to 0.2 s.
5. A cued double-modality switching task: Like stage 4 except that the single-modality trials were removed, and the block length was gradually shortened to three trials.

6. A selective attention task: Like stage 5, but the block structure was abandoned and all eight possible trial types were randomized: (audition vs vision) x (sound left or right) x (light left or right).

Iterative generalized linear model

We modeled the animal's choice probability by a logistic regression. At each trial number *t*, the choice probability is defined as

$$p\left(y_t = 1 \middle| \mathbf{w}_{t-1}\right) = \frac{1}{1+\exp\left(-\mathbf{w}_{t-1}^T \mathbf{x}_t\right)} \qquad (1)$$
$$p\left(y_t = -1 \middle| \mathbf{w}_{t-1}\right) = 1 - p\left(y_t = 1 \middle| \mathbf{w}_{t-1}\right)$$

where y_t indicates the binary choice of the animal (1 = right, –1 = left), $\mathbf{x}_t$ is the vector of input factors on trial *t*, and $\mathbf{w}_{t-1}$ is the vector of weights for these factors obtained from fitting up to the preceding trial.

The prediction y_t^* for the animal's choice is simply that with the higher model probability:

$$y_t^* = \underset{y \in \{-1,+1\}}{\arg\max}\ p\left(y \middle| \mathbf{x}_t, \mathbf{w}_{t-1}\right) \qquad (2)$$

After observing the animal's actual choice z_t, the cross-entropy error E_t between the observation and model prediction is calculated as

$$E_t = -\log p\left(z_t \middle| \mathbf{x}_t, \mathbf{w}_{t-1}\right) \qquad (3)$$

We weight the error term by a reward factor R_t, and apply exponential temporal smoothing to get the loss function L_t:

$$L_t = R_t E_t + \alpha L_{t-1} \qquad (4)$$

where α is the smoothing discount factor accounting for the effect that distant trials have less impact on decision-making than immediately preceding trials, and R_t is defined as

$$R_t = \begin{cases} 1, \text{if the choice is rewarded} \\ r, \text{otherwise} \end{cases} \qquad (5)$$

The values of R_t for rewarded and unrewarded trials may be different, accounting for the fact that rewards and punishments may have different effects on learning. For each time point, the weights in the model are determined by minimizing the loss function subject to L1 (lasso) regularization, namely

$$\mathbf{w}_t = \underset{\mathbf{w}}{\arg\min}\left(L_t + \lambda \|\mathbf{w}\|_1\right) \qquad (6)$$

Then $\mathbf{w}_t$ is used for prediction of the next trial. For subsequent analysis, we only used predictions starting at the 15th trial. The three hyperparameters for the temporal discount factor α, the reward factor r, and the regularization factor λ were selected by grid search.

To fit the decision-making sequences of the simple visual task, we included the following terms in the input vector $\mathbf{x}_t$:

1. Visual_stimulus: +1 = light on right, –1 = light on left.
2. Bias: A constant value of +1. The associated weight determines whether the animal favors the left (negative) or the right (positive) port.
3. Choice_back_n: The choice the animal made *n* trials ago (+1 = right, –1 = left).
4. Reward_back_n: The reward the animal received *n* trials ago (+1 = reward, –1 = punishment).
5. Choice x Reward_back_n: The product of terms 4 and 5. This term corresponds to the win-stay-lose-switch (WSLS) strategy of repeating the last choice if it was rewarded and switching if it was punished.

To determine the extent of history-dependence of the animal's decisions, we fitted the model including terms 3-5 from up to three previous trials ($n = 1, 2, 3$), and found that only the immediately preceding trial had an appreciable effect on the model's prediction accuracy. For the subsequent analysis, we therefore included terms 3-5 for the preceding trial ($n = 1$).

We compared the iterative generalized linear model (GLM) with two other models. The first only captures the animal's average performance over all trials. If the fraction of the correct responses is z, then the model simply predicts a correct response with probability z, and an error with probability $1–z$. Thus, the fraction of trials where the prediction matches the observation is $z^2 + \left(1-z\right)^2$.

The second model is a sliding window logistic regression. To make a prediction for trial t, we fitted the logistic model presented above (Eqns 1-2) to the preceding n trials. The loss function is

$$L_t = -\sum_{i=t-n}^{t-1} \log p\left(z_t \middle| \mathbf{x}_t, \mathbf{w}_{t-1}\right) \tag{7}$$

and the weights are again optimized as in Eqn 6.

Recovering policy matrices from simulated data

To test the model's ability in recovering policy matrices, we trained the model on data generated from pre-defined ground truth policies. The ground truth policies changed every 10 trials, 30 trials, or 90 trials. Binary choices were simulated with different noise levels using the algorithm 'epsilon-greedy': with a probability of epsilon, the simulator made a random choice and with a probability of 1-epsilon it chose the action indicated by the ground truth policy. The noise levels (epsilon values) ranged from 0 to 0.6. The similarity between the recovered policy and the ground truth policy was evaluated by the cosine between the recovered weight vector and the ground truth weight vector.

Automated behavior assessment software

In this section we describe each part of the software that constitutes the system we developed to track the mouse. The software is primarily made of two parts: mouse detection and pose estimation, each of which is implemented by a deep convolutional neural network (CNN) trained on annotated video data (**Supplementary Fig. 7a**). We collected a set of videos using red or IR light illumination from the top of the arena. From these videos we extracted randomly a set of 15,000 frames and asked Amazon Mechanical Turk (AMT) workers to click on body landmarks which give a representation of the skeleton of the mouse (**Supplementary Fig. 7b**).

Mouse detection

The architecture used for detection is the multi-scale convolutional (MSC) Multibox network[28], which computes a list of bounding boxes along with a single confidence score for each box, corresponding to its likelihood of containing the object of interest, in this case a mouse. Each bounding box is encoded by 4 scalars, representing the coordinates of the upper-left and lower-right coordinates. The coordinates of each box are normalized with respect to image dimensions to deal with different image sizes, and their associated confidence score is encoded by an additional node (which ouputs a value from 0 to 1). The loss function is the weighted sum of two losses: confidence and location[28]. We trained the MSC-Multibox deep CNN to predict bounding boxes that are spatially closest to the ground truth boxes while maximizing the confidence of containing the mouse.

In order to reach better and faster detection, we used prior bounding boxes whose aspect ratios closely match the distribution of the ground truth. As proposed previously by Erhan et al[28], these priors were selected because their Intersection over Union (IoU) with respect to the ground truth was over 0.5. In the following matching process, each ground truth box is best matched to a prior, and the algorithm learns a residual between the two. At inference time, 100 bounding boxes are proposed, from which the best one is selected based on the highest score and non-maximum suppression.

We split the dataset into 12,750 frames for training, 750 for validation and 1,500 for testing. During training, we augmented data with random cropping and color variation. Using the Inception-ResNet-V2 architecture initialized with ImageNet pre-trained weights[45], we finetuned the network with our training samples by updating the weights using stochastic gradient descent. For the optimizer, we used RMPSProp, with the batch size set to 4, the initial learning rate set to 0.01, and the momentum and the decay both set to 0.99[46]. Images were resized to 299 x 299. We trained the detector on a machine with a 8-core Intel i7-6700K CPU, 32GB of RAM, and a 8GB GTX 1080 GPU. The model was trained for 288k iterations. A single instance of the forward pass took on average 15 ms.

We evaluated the models using the detection metric Intersection over Union (IoU). Thresholding the IoU defines matches between the ground truth and predicted boxes and allows computing precision-recall curves. The precision-recall curve at different threshold of IoU is shown in **Supplementary Fig. 7d**. In **Supplementary Table 2**, we report mean averaged precision (mAP) and recall (mAR).

Pose estimation

With the bounding box generated from the MSC-Multibox deep CNN, we wish to determine the precise pixel location of the keypoints that would describe the body features of the mouse. As a well established problem in computer vision, a good pose estimation system must be robust to occlusion, deformation, successful on rare and novel poses, invariant to changes in appearance from differences in lighting and backgrounds.

The keypoints we chose are the nose, the ears, the neck, the body sides, and the base of the tail (**Supplementary Fig. 7b**). These features were chosen because they are best recognized regardless of the size of the animal, and one can deduce from them secondary features, such as orientation of the animal. We used Stacked Hourglass Network[29] to estimate the keypoints. This architecture has the capacity to learn all seven features and and output pixel-level predictions. The output of the network is a set of heatmaps, one for each keypoint, representing the probability of the keypoint's presence every pixel (**Supplementary Fig. 7a**). We estimated the location of the keypoint by the maximum of its heatmap. A mean squared error (MSE) loss was used to compare the predicted heatmap to the ground truth.

During training, cropped frames with the mouse centered in the bounding box were resized to a resolution of 256 x 256. We also augmented the training data as follows (p is the probability of applying a type of augmentation):

- rotation with $p = 1$: angles were selected uniformly between 0° and 180°
- translation
- horizontal and vertical flips
- scaling with $p = 1$: scaling factors were chosen from a pool of 0.10 to 0.65, uniformly
- color variation: adjusted brightness/contrast/gamma with $p = 0.5$ in order to emulate the effects of poor lighting/setup
- Gaussian blur with $p = 0.15$: frames were blurred either by a $\sigma = 1$ or $\sigma = 2$ (chosen uniformly).
- Gaussian Noise added independently across image with $p = 0.15$
- JPEG artifact with $p = 0.15$: added artifacts of JPEG compression onto the image

Extreme augmentations (with multiple types of augmentations) were examined to make sure that the transformed data looked reasonable. Using original and augmented keypoint annotations, we trained a pose estimator from scratch.

Training started from randomly initialized weights, and continued until validation accuracy plateaued, taking approximately 6 days. This training process was performed for 749k iterations. The network was trained using TensorFlow (Google) on a machine with 8-core Intel Xeon CPU, 24GB of RAM, and a 12GB Titan XP GPU. For optimization, we used RMPSProp optimizer with momentum and decay both set to 0.99, batch size of 8 and a learning rate of 0.00025. We dropped the learning rate once by a factor of 5 after validation accuracy plateaus (after 33 epochs). Batch normalization was used to improve training.

Evaluation was done using the standard Percentage of Correct Keypoint (PCK) metric which reports the fraction of detections that fall within a distance of the ground truth[29]. More than 85% of the keypoints of the nose, ears, and neck are inferred within an error radius of 0.5 cm, and more than 80% of the keypoints of the body sides and tail lie within an error error radius of 1 cm (as a reference, the distance between two ears is ~3 cm). The averaged PCK of all the seven keypoints is ~80% within a radius of less than 0.5 cm (**Supplementary Fig. 7c**). Overall the system's performance can be characterized as high-human, significantly exceeding the typical annotator, but less precise than the absolute best possible.

Supervised and unsupervised analysis of behavioral trajectories

From the pose estimation, we extracted two features to describe an animal's behavioral trajectories: the centroid was defined as the average position of the seven body landmarks; and the angular orientation of the line from the centroid to the nose. For each trial, these two features were extracted for n frames (n = 30 (1 s) in most cases), thus the data dimension for each trial is $3n$ (the two centroid coordinates and the orientation).

To determine whether the behavioral trajectories contain information about the decision categories, a support vector machine (SVM) with a linear kernel was trained for each decision category. The training set was labelled with the decision category based on information about the visual stimulus and the animal's choice (for example, "Stim: R, Choice: L" means that the light is on the right and the animal chooses the left port). Performance of the trained SVM was examined by prediction accuracy on the test set, and the F1 score, which is the harmonic mean of precision and recall:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} = \frac{2 \cdot true\_positive}{2 \cdot true\_positive + false\_positive + false\_negative} \quad (8)$$

The performance was computed as the average across 10 repeated analyses (**Supplementary Fig. 8**).

We performed a non-linear embedding method, t-distributed stochastic neighbor embedding (t-SNE) analysis as previously described[31,32]. Briefly, the trajectory data of each trial were projected into a 2D t-SNE space. Point clouds on the t-SNE map represented candidate clusters. Density clustering identified these regions. We then plotted trajectories and reaction time distributions to confirm that the clusters were distinct from each other. A repository of the analysis scripts can be found in https://github.com/tonyzhang25/MouseAcademyBehavior.

AUTHOR CONTRIBUTIONS

M.Q. and M.M. designed the study; M.Q. and S.S. constructed the hardware setup and wrote the controlling software; M.Q. performed experiments and collected data for analysis; M.Q. developed the iterative generalized linear model with input from P.P. and M.M; C.S. developed the automated tracking software; T.Z. implemented animal tracking and behavioral trajectory analysis with input from M.Q., P.P. and M.M; M.Q. and M.M. wrote the manuscript with comments from all authors.

ACKNOWLEDGEMENT

We thank Joshua Sanders for technical assistance in incorporating Bpod into our system. We thank Ann Kennedy for insightful comments and suggestions on analysis of the behavioral trajectories. We thank Oisin Mac Adoha and Yuxin Chen for helpful comments and discussions. This work was supported by a grant from the Simons Foundation (SCGB 543015, M.M. and P.P.) and a postdoctoral fellowship from the Swartz Foundation (M.Q.).

COMPETING FINANCIAL INTERESTS

The authors declare no competing financial interests.

DATA AVAILABILITY

The datasets analyzed during the current study are available in https://drive.google.com/open?id=1gkPbqGYKPGs7Rx1WNmubQW0dKyYE5YVR

REFERENCES

1 Spear, N. E., Miller, J. S. & Jagielo, J. A. Animal memory and learning. *Annu Rev Psychol* **41**, 169-211, doi:10.1146/annurev.ps.41.020190.001125 (1990).

2 Branson, K., Robie, A. A., Bender, J., Perona, P. & Dickinson, M. H. High-throughput ethomics in large groups of Drosophila. *Nature methods* **6**, 451-457, doi:10.1038/nmeth.1328 (2009).

3 Honegger, K. & de Bivort, B. Stochasticity, individuality and behavior. *Current biology : CB* **28**, R8-R12, doi:10.1016/j.cub.2017.11.058 (2018).

4 Matzel, L. D. & Sauce, B. Individual differences: Case studies of rodent and primate intelligence. *J Exp Psychol Anim Learn Cogn* **43**, 325-340, doi:10.1037/xan0000152 (2017).

5 Atkinson, J. W. & Raphelson, A. C. Individual differences in motivation and behavior in particular situations. *J Pers* **24**, 349-363 (1956).

6 Stern, E. Individual differences in the learning potential of human beings. *Npj Science of Learning* **2** (2017).

7 Luo, L., Callaway, E. M. & Svoboda, K. Genetic Dissection of Neural Circuits: A Decade of Progress. *Neuron* **98**, 865, doi:10.1016/j.neuron.2018.05.004 (2018).

8 Carandini, M. & Churchland, A. K. Probing perceptual decisions in rodents. *Nature neuroscience* **16**, 824-831, doi:10.1038/nn.3410 (2013).

9 Gomez-Marin, A., Paton, J. J., Kampff, A. R., Costa, R. M. & Mainen, Z. F. Big behavioral data: psychology, ethology and the foundations of neuroscience. *Nature neuroscience* **17**, 1455-1462, doi:10.1038/nn.3812 (2014).

10 Juavinett, A. L., Erlich, J. C. & Churchland, A. K. Decision-making behaviors: weighing ethology, complexity, and sensorimotor compatibility. *Current opinion in neurobiology* **49**, 42-50, doi:10.1016/j.conb.2017.11.001 (2018).

11 Schmitt, L. I. *et al.* Thalamic amplification of cortical connectivity sustains attentional control. *Nature* **545**, 219-223, doi:10.1038/nature22073 (2017).

12 Wimmer, R. D. *et al.* Thalamic control of sensory selection in divided attention. *Nature* **526**, 705-709, doi:10.1038/nature15398 (2015).

13 Akrami, A., Kopec, C. D., Diamond, M. E. & Brody, C. D. Posterior parietal cortex represents sensory history and mediates its effects on behaviour. *Nature* **554**, 368-372, doi:10.1038/nature25510 (2018).

14 Uchida, N. & Mainen, Z. F. Speed and accuracy of olfactory discrimination in the rat. *Nature neuroscience* **6**, 1224-1229, doi:10.1038/nn1142 (2003).

15 Scott, B. B., Brody, C. D. & Tank, D. W. Cellular resolution functional imaging in behaving rats using voluntary head restraint. *Neuron* **80**, 371-384, doi:10.1016/j.neuron.2013.08.002 (2013).

16 Busse, L. *et al.* The detection of visual contrast in the behaving mouse. *The Journal of neuroscience : the official journal of the Society for Neuroscience* **31**, 11351-11361, doi:10.1523/JNEUROSCI.6689-10.2011 (2011).

17 Poddar, R., Kawai, R. & Olveczky, B. P. A Fully Automated High-Throughput Training System for Rodents. *PloS one* **8**, doi:ARTN e83171
10.1371/journal.pone.0083171 (2013).

18 Silasi, G. *et al.* Individualized tracking of self-directed motor learning in group-housed mice performing a skilled lever positioning task in the home cage. *Journal of Neurophysiology* **119**, 337-346, doi:10.1152/jn.00115.2017 (2018).

19 Erskine, A. B., T.; Herb, J. T.; Schaefer, A. AutonoMouse: High throughput automated operant conditioning shows progressive behavioural impairment with graded olfactory bulb lesions. https://www.biorxiv.org/content/early/2018/03/29/291815 (2018).

20 Crabbe, J. C., Wahlsten, D. & Dudek, B. C. Genetics of mouse behavior: interactions with laboratory environment. *Science* **284**, 1670-1672 (1999).

21 Hurst, J. L. & West, R. S. Taming anxiety in laboratory mice. *Nature methods* **7**, 825-826, doi:10.1038/nmeth.1500 (2010).

22 Daw, N. D. in *Decision Making, Affect, and Learning: Attention and Performance XXIII* (Oxford University Press, 2011).

23 Winter, Y. & Schaefers, A. T. A sorting system with automated gates permits individual operant experiments with mice from a social home cage. *J Neurosci Methods* **196**, 276-280, doi:10.1016/j.jneumeth.2011.01.017 (2011).

24 Daan, S. *et al.* Lab mice in the field: unorthodox daily activity and effects of a dysfunctional circadian clock allele. *J Biol Rhythms* **26**, 118-129, doi:10.1177/0748730410397645 (2011).

25 Hut, R. A., Pilorz, V., Boerema, A. S., Strijkstra, A. M. & Daan, S. Working for food shifts nocturnal mouse activity into the day. *PloS one* **6**, e17527, doi:10.1371/journal.pone.0017527 (2011).

26 Seidemann, E. Neuronal mechanisms mediating conversion of visual signals into perceptual decisions in a direction discrimination task. *Ph.D. dissertation, Stanford University.* (1998).

27 Burgess, C. P. *et al.* High-Yield Methods for Accurate Two-Alternative Visual Psychophysics in Head-Fixed Mice. *Cell Rep* **20**, 2513-2524, doi:10.1016/j.celrep.2017.08.047 (2017).

28 Erhan, D. S., D.; Toshev, A.; Anguelov, D. Scalable Object Detection using Deep Neural Networks. https://arxiv.org/abs/1312.2249 (2014).

29 Newell, A. Y., K.; Deng, J. Stacked Hourglass Networks for Human Pose Estimation. https://arxiv.org/abs/1603.06937 (2016).

30 Gris, K. V., Coutu, J. P. & Gris, D. Supervised and Unsupervised Learning Technology in the Study of Rodent Behavior. *Front Behav Neurosci* **11**, 141, doi:10.3389/fnbeh.2017.00141 (2017).

31 Berman, G. J., Choi, D. M., Bialek, W. & Shaevitz, J. W. Mapping the stereotyped behaviour of freely moving fruit flies. *J R Soc Interface* **11**, doi:10.1098/rsif.2014.0672 (2014).

32 Todd, J. G., Kain, J. S. & de Bivort, B. L. Systematic exploration of unsupervised methods for mapping behavior. *Phys Biol* **14**, 015002, doi:10.1088/1478-3975/14/1/015002 (2017).

33 Botvinick, M. M., Braver, T. S., Barch, D. M., Carter, C. S. & Cohen, J. D. Conflict monitoring and cognitive control. *Psychol Rev* **108**, 624-652 (2001).

34 Sorge, R. E. *et al.* Olfactory exposure to males, including men, causes stress and related analgesia in rodents. *Nature methods* **11**, 629-632, doi:10.1038/nmeth.2935 (2014).

35 Barnes, C. A. Memory deficits associated with senescence: a neurophysiological and behavioral study in the rat. *J Comp Physiol Psychol* **93**, 74-104 (1979).

36 Olton, D. S., Walker, J. A. & Gage, F. H. Hippocampal connections and spatial discrimination. *Brain Res* **139**, 295-308 (1978).

37 Aoki, R., Tsubota, T., Goya, Y. & Benucci, A. An automated platform for high-throughput mouse behavior and physiology with voluntary head-fixation. *Nat Commun* **8**, 1196, doi:10.1038/s41467-017-01371-0 (2017).

38 Shalev-Shwartz, S. Online Learning and Online Convex Optimization. *Foundations and Trends in Machine Learning* **4**, 107-194 (2012).

39 Piech, C. S., J.; Huang, J.; Ganguli, S.; Sahami, M.; Guibas, L.; Sohl-Dickstein, J. Deep Knowledge Tracing. https://arxiv.org/abs/1506.05908 (2015).

40 Hong, W. *et al.* Automated measurement of mouse social behaviors using depth sensing, video tracking, and machine learning. *Proc Natl Acad Sci U S A* **112**, E5351-5360, doi:10.1073/pnas.1515982112 (2015).

41 Egnor, S. E. & Branson, K. Computational Analysis of Behavior. *Annu Rev Neurosci* **39**, 217-236, doi:10.1146/annurev-neuro-070815-013845 (2016).

42 Wiltschko, A. B. *et al.* Mapping Sub-Second Structure in Mouse Behavior. *Neuron* **88**, 1121-1135, doi:10.1016/j.neuron.2015.11.031 (2015).

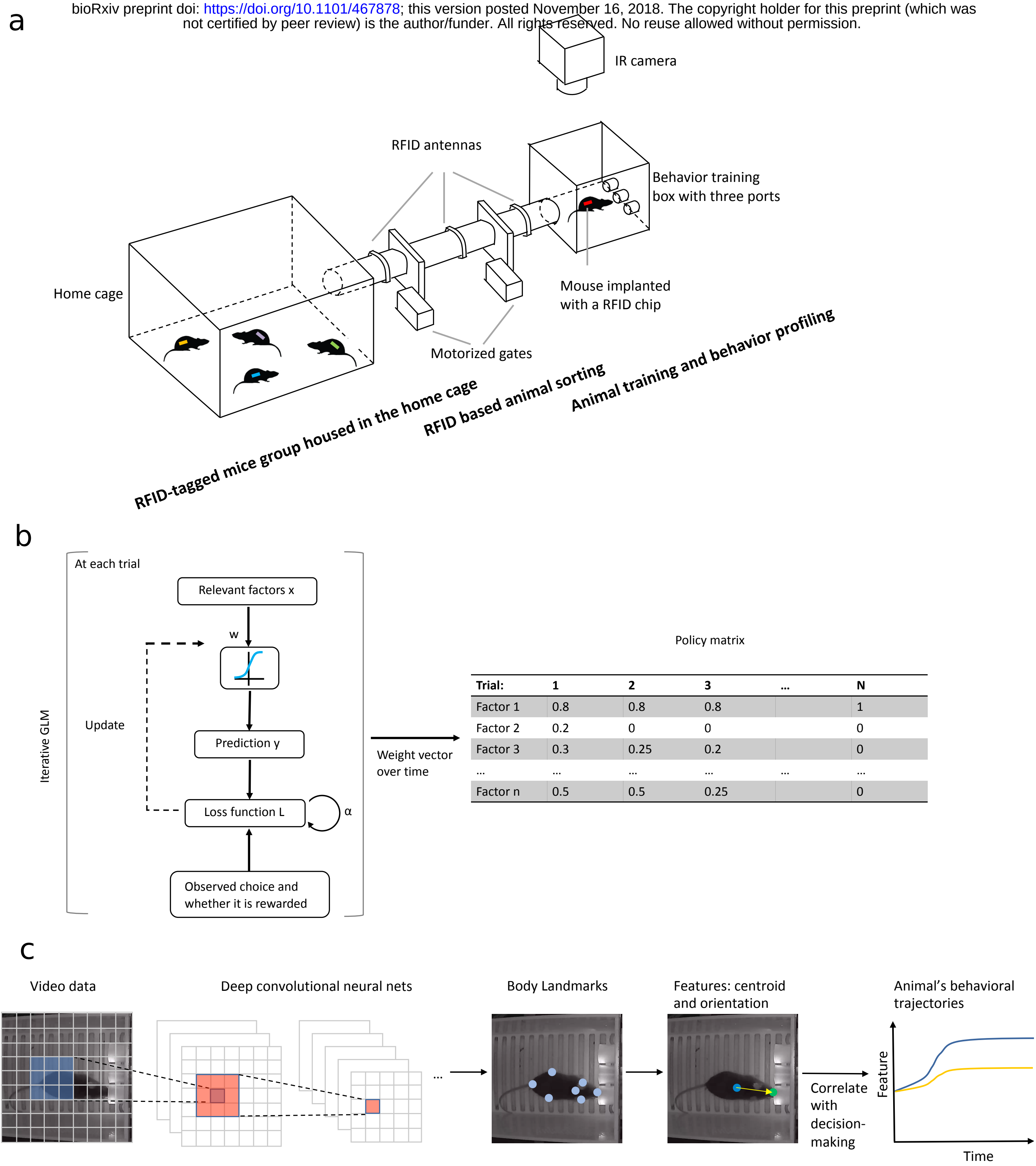
43 Szuts, T. A. *et al.* A wireless multi-channel neural amplifier for freely moving animals. *Nature neuroscience* **14**, 263-269, doi:10.1038/nn.2730 (2011).

44 Ziv, Y. *et al.* Long-term dynamics of CA1 hippocampal place codes. *Nature neuroscience* **16**, 264-266, doi:10.1038/nn.3329 (2013).

45 Szegedy, S. I., S.; Vanhoucke, V.; Alemi, A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. https://arxiv.org/abs/1602.07261 (2016).

46 Ruder, S. An overview of gradient descent optimization algorithms. https://arxiv.org/abs/1609.04747 (2017).

Figure 1

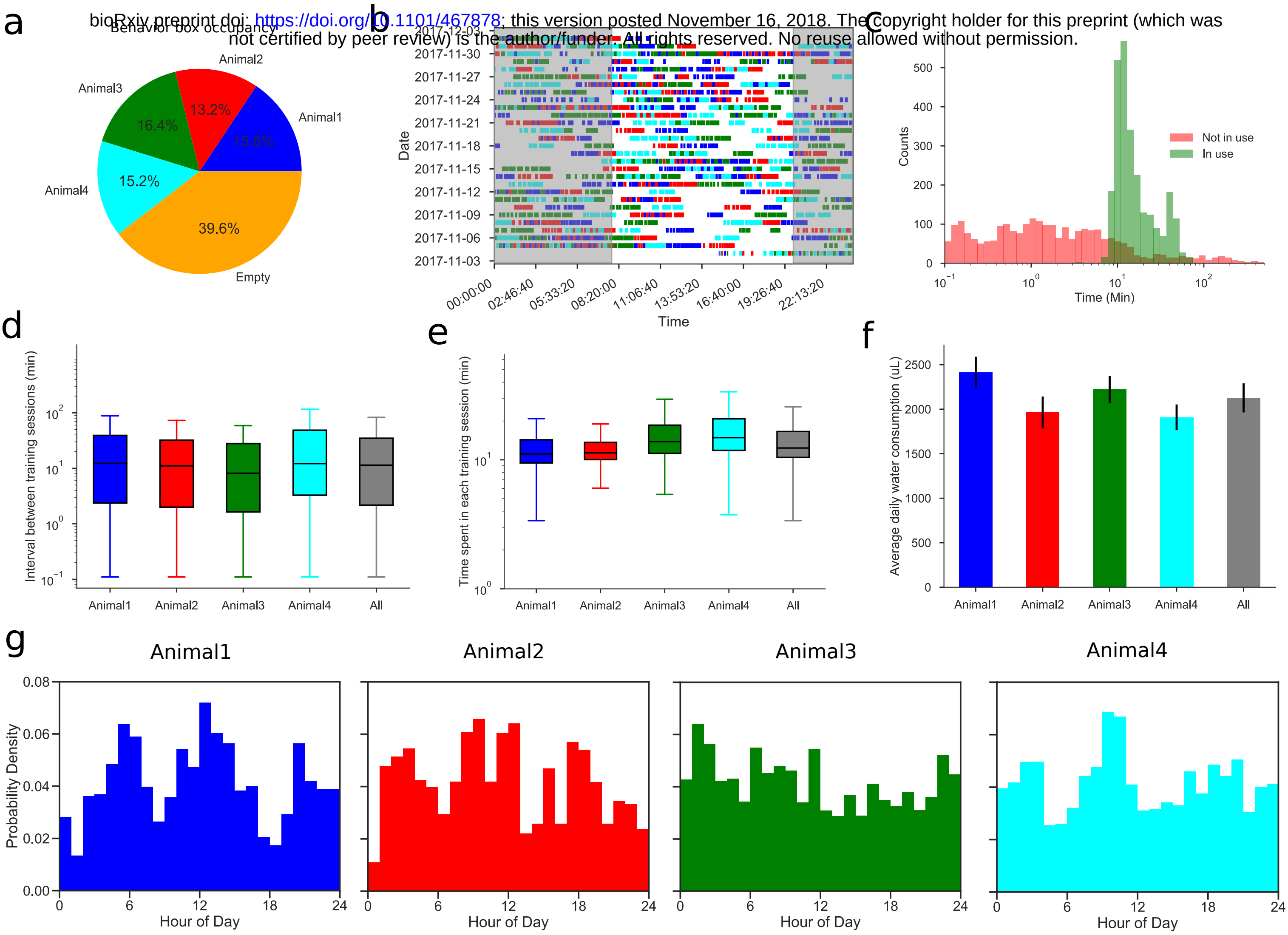


Trial:	1	2	3	...	N
Factor 1	0.8	0.8	0.8		1
Factor 2	0.2	0	0		0
Factor 3	0.3	0.25	0.2		0
...	...	...	...	...	...
Factor n	0.5	0.5	0.25		0

Fig. 1: Components of Mouse Academy

(a) An automated RFID sorting and animal training system. Mice implanted with RFID chips are group-housed in the home cage. The RFID sorting system identifies each mouse by its implanted chip. One animal at a time gains access to a behavioral training box. As the animal learns a task, its decision sequences and video recordings are acquired. **(b)** An iterative generalized linear model. For each trial, the model predicts the animal's choice based on the relevant factors and then evaluates the difference from the actual choice. This difference, after temporal weighting, is fed back to the loss function, which gets minimized by updating the weights of the input factors. The model produces a policy matrix in which the rows indicate the weights of the relevant factors and the columns are the trials. **(c)** An automated behavior assessment program using deep convolutional neural networks to extract the location and pose information of an animal.

Figure 2

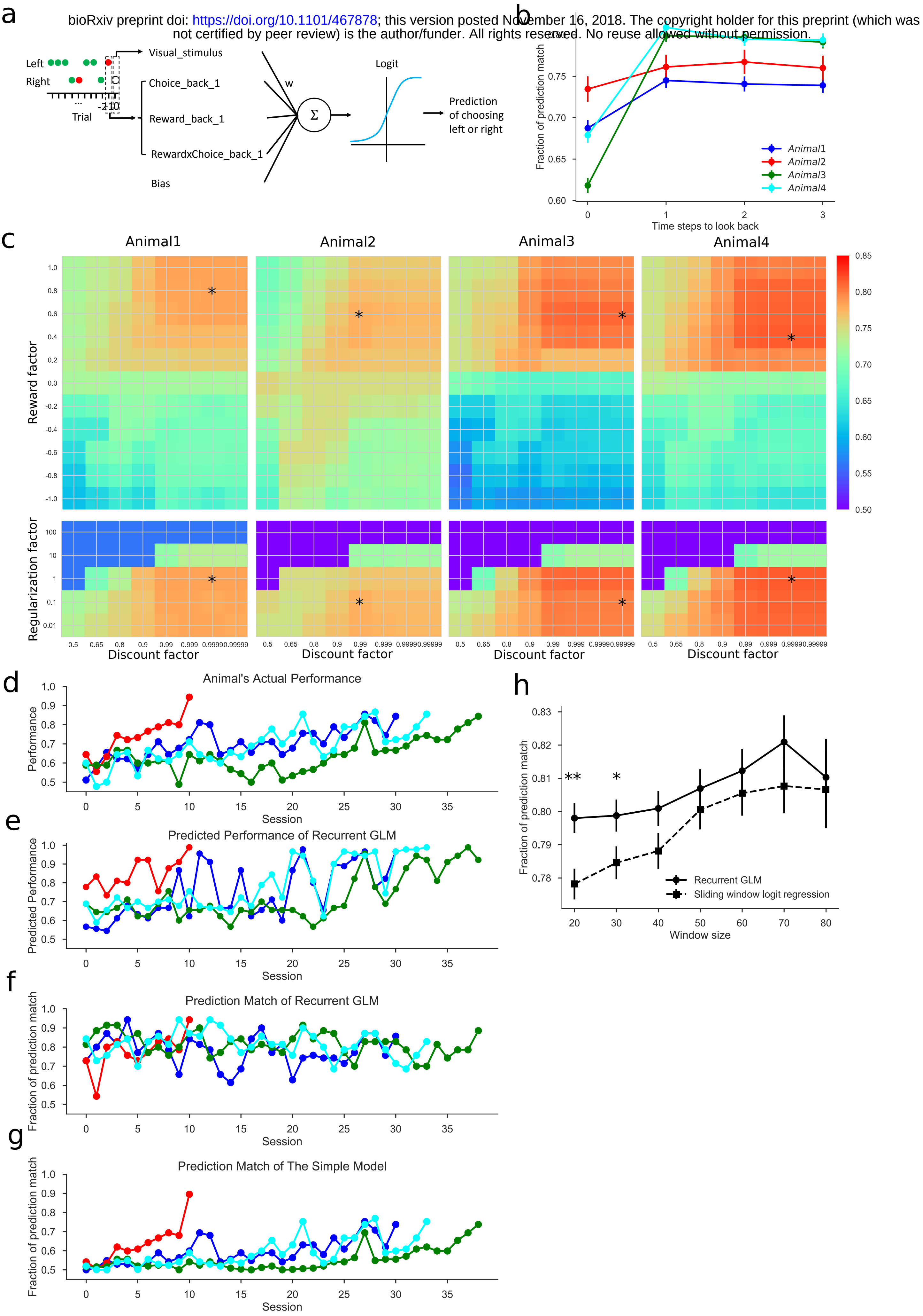


a
Behavior box occupancy
Animal1
Animal2
Animal3
Animal4
Empty
15.6%
13.2%
16.4%
15.2%
39.6%
b
Date
Time
2017-12-03
2017-11-30
2017-11-27
2017-11-24
2017-11-21
2017-11-18
2017-11-15
2017-11-12
2017-11-09
2017-11-06
2017-11-03
00:00:00
02:46:40
05:33:20
08:20:00
11:06:40
13:53:20
16:40:00
19:26:40
22:13:20
c
Counts
Time (Min)
Not in use
In use
d
Interval between training sessions (min)
e
Time spent in each training session (min)
f
Average daily water consumption (uL)
Animal1
Animal2
Animal3
Animal4
All
g
Probability Density
Hour of Day
Animal1
Animal2
Animal3
Animal4

Fig. 2: Performance of the automated training system on a sample cohort

(a) Fraction of time the behavior box was occupied by each of the four animals. **(b)** Activity trace of each animal in the behavior box for the entire training period of 28 days. Shadow indicates the dark cycle from 8pm to 8am. **(c)** Distribution of time intervals during which the behavior box is occupied or empty. **(d)** Box plot of intervals between each animal's sessions (median, quartiles, and range). **(e)** Box plot of the time spent in a session for each animal. **(f)** Averaged daily water consumption of each animal. Error bars indicate standard errors. **(g)** Circadian histograms of each animal's activity in the behavior box.

Figure 3



a
Left
Right
Trial
-2 -1 0
Visual_stimulus
Choice_back_1
Reward_back_1
RewardxChoice_back_1
Bias
w
Σ
Logit
Prediction of choosing left or right
b
Fraction of prediction match
Time steps to look back
Animal1
Animal2
Animal3
Animal4
c
Animal1
Animal2
Animal3
Animal4
Reward factor
Regularization factor
Discount factor
d
Animal's Actual Performance
Performance
Session
e
Predicted Performance of Recurrent GLM
Predicted Performance
Session
f
Prediction Match of Recurrent GLM
Fraction of prediction match
Session
g
Prediction Match of The Simple Model
Fraction of prediction match
Session
h
Fraction of prediction match
Window size
Recurrent GLM
Sliding window logit regression

Fig. 3: Iterative generalized linear model and its prediction accuracy
(a) Illustration of the GLM as applied to a visual discrimination task. The model's prediction is based on the output of a logistic function whose input is the weighted sum of a visual stimulus term, a bias term, and three history-dependent terms. The stimulus can be on the left or right and the choice can be rewarded (consistent with the stimulus, indicated by a green dot) or unrewarded (opposite to the stimulus, indicated by a red dot). **(b)** Selection of the history-dependent terms based on the model prediction accuracy. Error bars indicate standard errors. **(c)** Hyperparameters for each of the animals: reward factor, discount factor, and regularization factor. The optimal values are marked with a star. **(d)** The actual performance of each animal over time in the visual task. **(e)** Performance as predicted by the GLM. **(f)** Fraction of choices predicted correctly by the GLM. **(g)** Fraction of choices predicted correctly by a simple model based on the animal's average performance in the task. **(h)** Fraction of predictions matched by the iterative GLM and the sliding window logistic regression model. Error bars indicate standard errors. **, * indicate $P < 0.01, 0.05$. Random prediction would give 50% match.

Figure 4

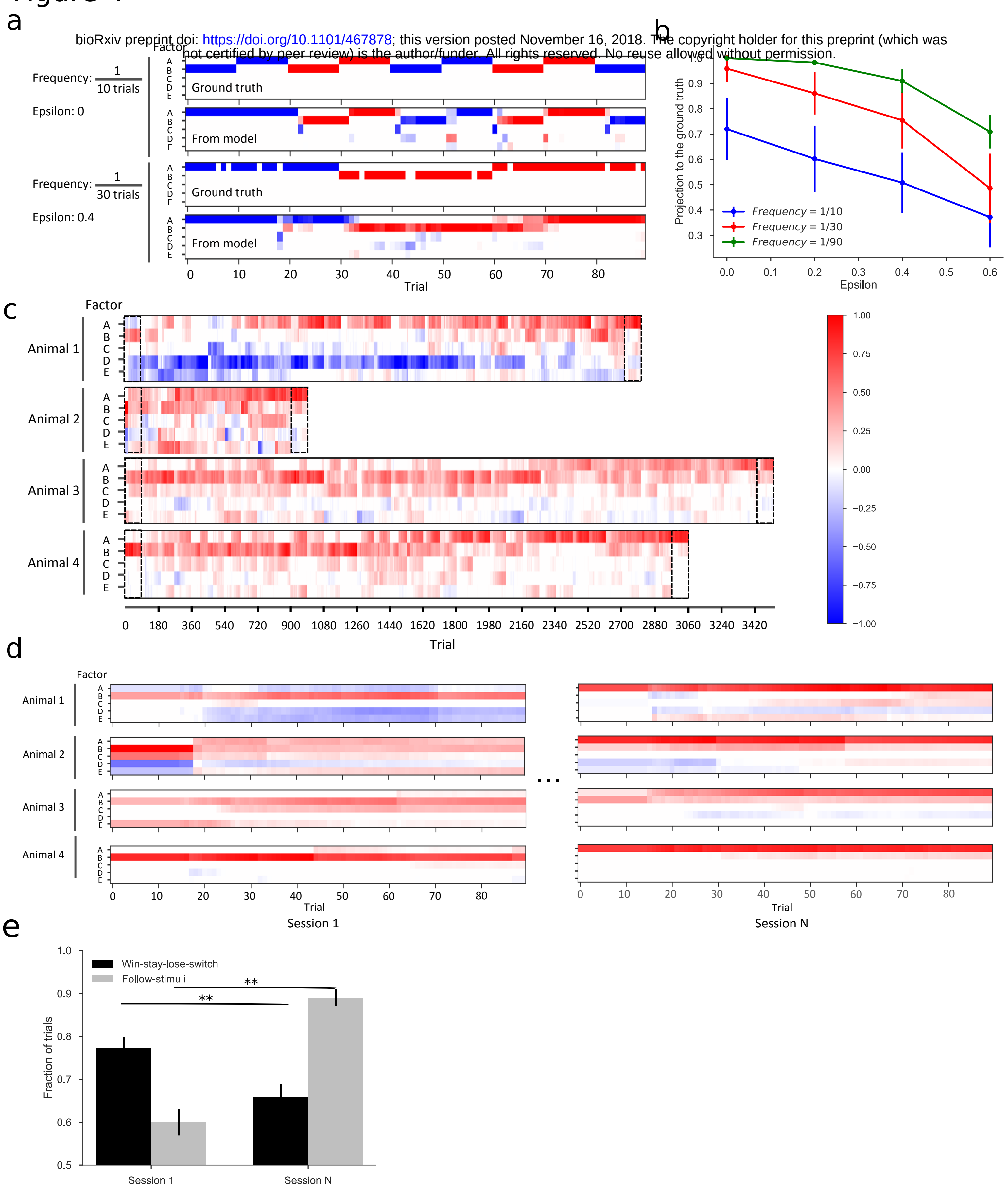

a
Factor
Frequency: 1/10 trials
Epsilon: 0
Ground truth
From model
Frequency: 1/30 trials
Epsilon: 0.4
Ground truth
From model
A
B
C
D
E
Trial
b
Projection to the ground truth
Epsilon
Frequency = 1/10
Frequency = 1/30
Frequency = 1/90
c
Factor
Animal 1
Animal 2
Animal 3
Animal 4
Trial
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
d
Factor
Animal 1
Animal 2
Animal 3
Animal 4
Trial
Session 1
Session N
e
Win-stay-lose-switch
Follow-stimuli
Fraction of trials
**
**
Session 1
Session N

Fig. 4: Interpretation of policies during learning

(a) Policy vectors recovered by the iterative GLM capture the ground truth policies. The policy matrix plots in each trial (horizontal) the weights associated with each of 5 factors (vertical), encoded with a color scale (see panel c). The factors are: A = Visual_stimulus, B = Choice×Reward_back_1, C = Choice_back_1, D = Reward_back_1, E = Bias. Two examples are shown of ground truth policies used to simulate data and the corresponding trial-by-trial estimates from the GLM. Blanks in the ground truth matrix indicate instances where the simulated choice is opposite to the policy. **(b)** Similarity between the recovered policy and the ground truth, measured by the cosine between the two policy vectors. Error bars indicate standard deviation. **(c)** Policy matrices recovered for the four animals show distinct individual learning processes. Dashed rectangles highlight the first and last sessions of each animal, as enlarged in **d**. **(d)** Recovered policy matrices for the first and last sessions of each animal. **(e)** Fraction of trials explained by two candidate policies (win-stay-lose-switch and following the stimuli) in the first and last sessions. Error bars indicate standard errors. ** indicates $P < 0.01$.

Figure 5

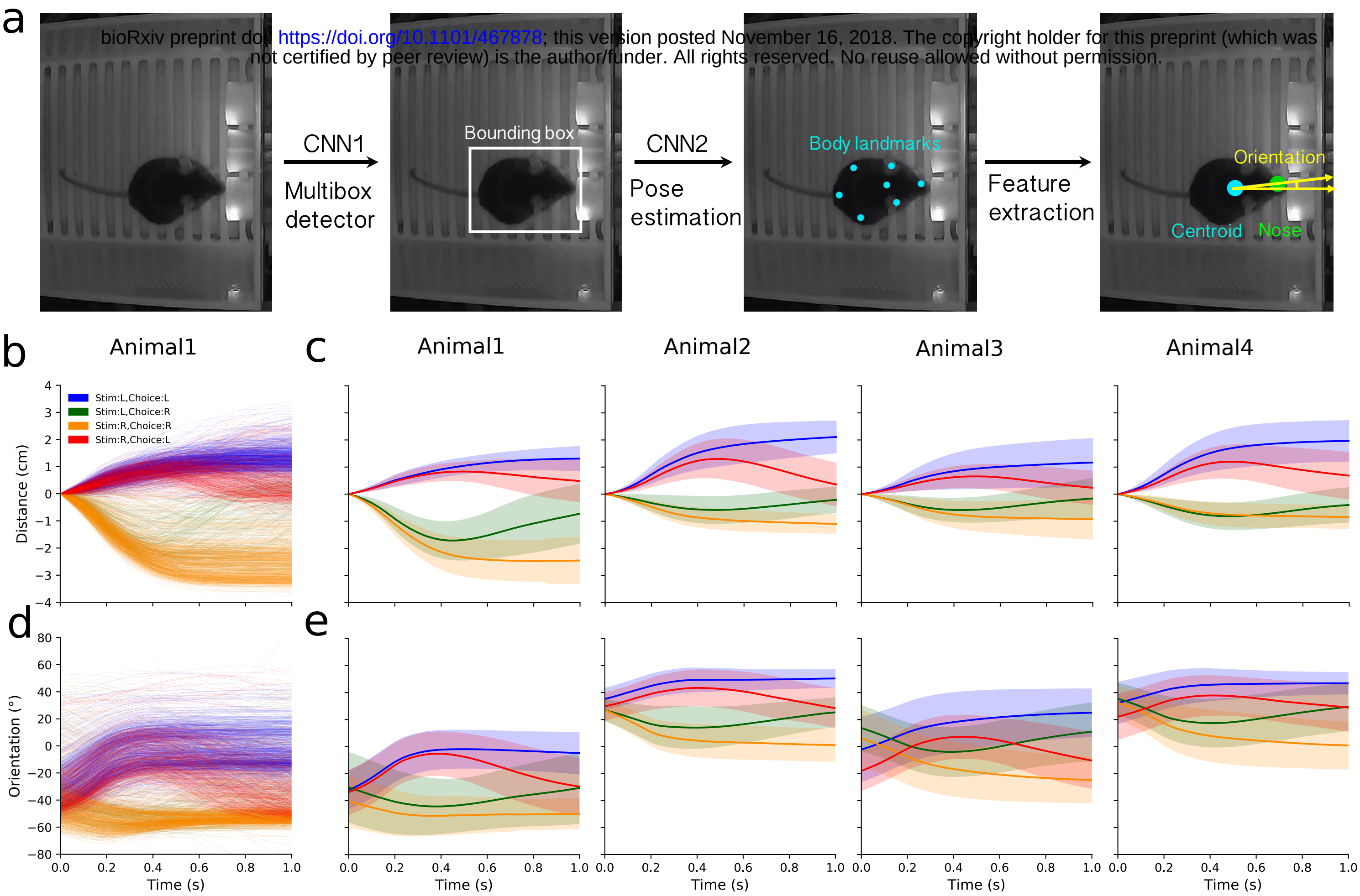

a
CNN1
Multibox detector
Bounding box
CNN2
Pose estimation
Body landmarks
Feature extraction
Orientation
Centroid
Nose
b
Animal1
Stim:L,Choice:L
Stim:L,Choice:R
Stim:R,Choice:R
Stim:R,Choice:L
Distance (cm)
c
Animal1
Animal2
Animal3
Animal4
d
Orientation (°)
e
Time (s)

Fig. 5: Supervised analysis using features extracted by automated behavior assessment
(a) Two deep convolutional neural networks (CNNs) extract relevant features of mouse behavior. The first CNN generates for each video frame a bounding box that encloses the animal. The second CNN operates on the bounding box and generates seven body landmarks for nose, ears, neck, body sides, and tail. The centroid is the average of the seven body landmarks and the orientation is the angle between the horizontal axis and the line connecting the centroid and the nose. **(b)** Centroid distance along the left-right axis vs time during the movement, for animal 1. The starting position is set to zero, positive values indicate movement to the left, negative to the right. The 4 trial types are indicated by different colors. **(c)** Average centroid trajectory for each animal. Shaded region indicates standard error. **(d-e)** Orientation vs time, displayed as in panels b-c. Positive angle points to the left, negative to the right.

Figure 6

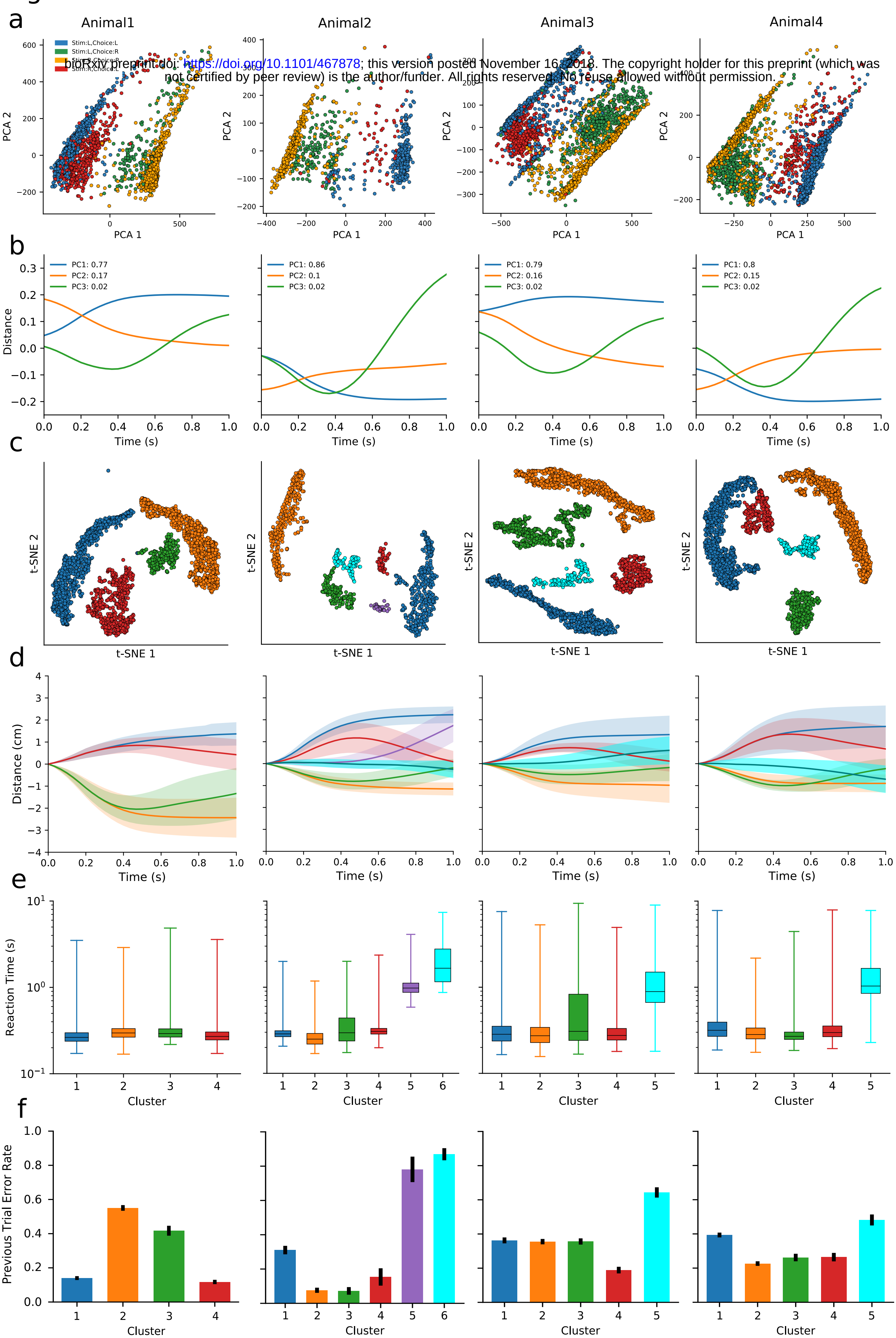

a
Animal1
Animal2
Animal3
Animal4
Stim:L,Choice:L
Stim:L,Choice:R
Stim:R,Choice:R
Stim:R,Choice:L
PCA 1
PCA 2
b
PC1: 0.77
PC2: 0.17
PC3: 0.02
PC1: 0.86
PC2: 0.1
PC3: 0.02
PC1: 0.79
PC2: 0.16
PC3: 0.02
PC1: 0.8
PC2: 0.15
PC3: 0.02
Distance
Time (s)
c
t-SNE 1
t-SNE 2
d
Distance (cm)
Time (s)
e
Reaction Time (s)
Cluster
f
Previous Trial Error Rate
Cluster

Fig. 6: Unsupervised analysis of the behavior trajectories

(a) Principal component projections onto PC1 and PC2 of the centroid-vs-time trajectories from Fig 5. The 4 trial types are indicated by different colors. **(b)** The centroid trajectories corresponding to the first three principal components (PCs). The variance explained by each PC is shown in the plot legend. **(c)** Clustering trials by their trajectories using t-SNE analysis. Distinct clusters are marked with different colors for use in subsequent panels. **(d)** Averaged centroid distance vs time for each cluster, plotted as in Fig 5b. **(e)** Box plot of the reaction time for each cluster. **(f)** The error rate on the preceding trial for each cluster. Error bars indicate standard errors.

Supplementary Materials

Supplementary Table 1. Cost Analysis of Mouse Academy Hardware

PC version

Parts	**Quantity**	**Supplier**	**Cost**
Bpod State Machine	1	Sanworks	$495.00
Custom behavior box (Plastic, ports with LEDs and photo-gates, valves, etc.)	1	Port breakout boards from Sanworks	$500.00
IR Webcam	1	Ailipu Technology	$47.99
Custom light- and sound-proof chamber	1	NA	$75.00
Arduino Mega 2560 R3	1	Sparkfun	$45.95
RFID Reader ID-12LA	3	Sparkfun	$29.95
Servo - Generic	2	Sparkfun	$8.95
Custom tunnel (Plastic, holders, antennas etc.)	1	NA	$150.00
Engineered home cage	1	Caltech animal facility	$76.10
PC	1	Hewlett-Packard	$1000.00
Total cost			$2428.94

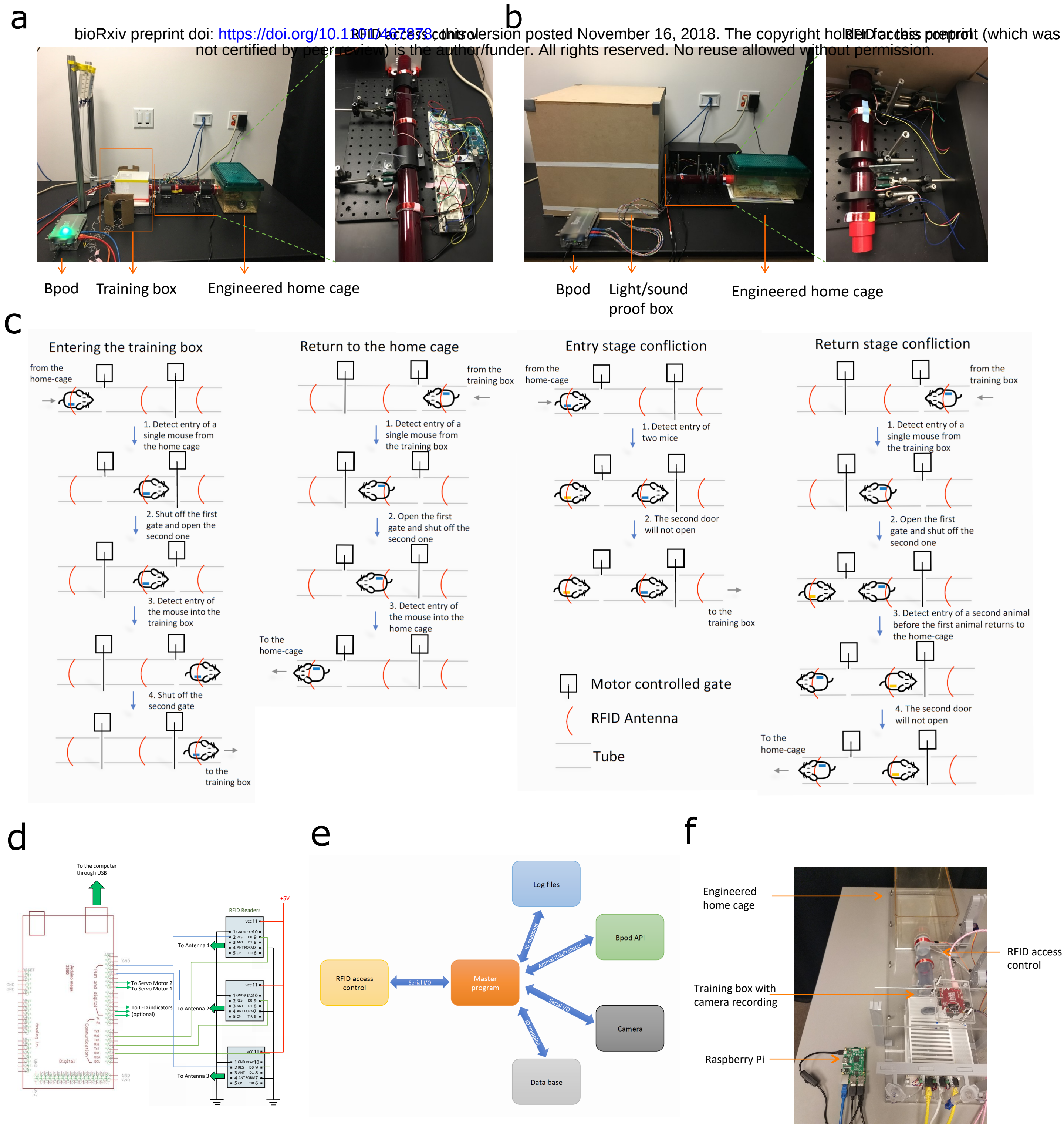
Raspberry Pi version

Parts	**Quantity**	**Supplier**	**Cost**
Bpod State Machine	1	Sanworks	$495.00
Custom behavior box (Plastic, ports with LEDs and photo-gates, valves, etc.)	1	Port breakout boards from Sanworks	$500.00
OpenMV Camera M7	1	OpenMV	$65.00
Custom light- and sound-proof chamber	1	NA	$75.00
Arduino Mega 2560 R3	1	Sparkfun	$45.95
RFID Reader ID-12LA	3	Sparkfun	$29.95
Servo - Generic	2	Sparkfun	$8.95
Custom tunnel (Plastic, holders, antennas etc.)	1	NA	$150.00
Engineered home cage	1	Caltech animal facility	$76.10
Raspberry Pi 3 Model B	1	Raspberry Pi	$35.10
Total cost			$1481.05

Supplementary Table 2: mean averaged precision (mAP) and recall (mAR) at different threshold of Intersection over Union (IoU) (related to mouse detection)

	mAP	mAR
$0.75 < IoU < 0.95$	0.81	0.82
$IoU = 0.5$	0.99	0.99
$IoU = 0.75$	0.95	0.98

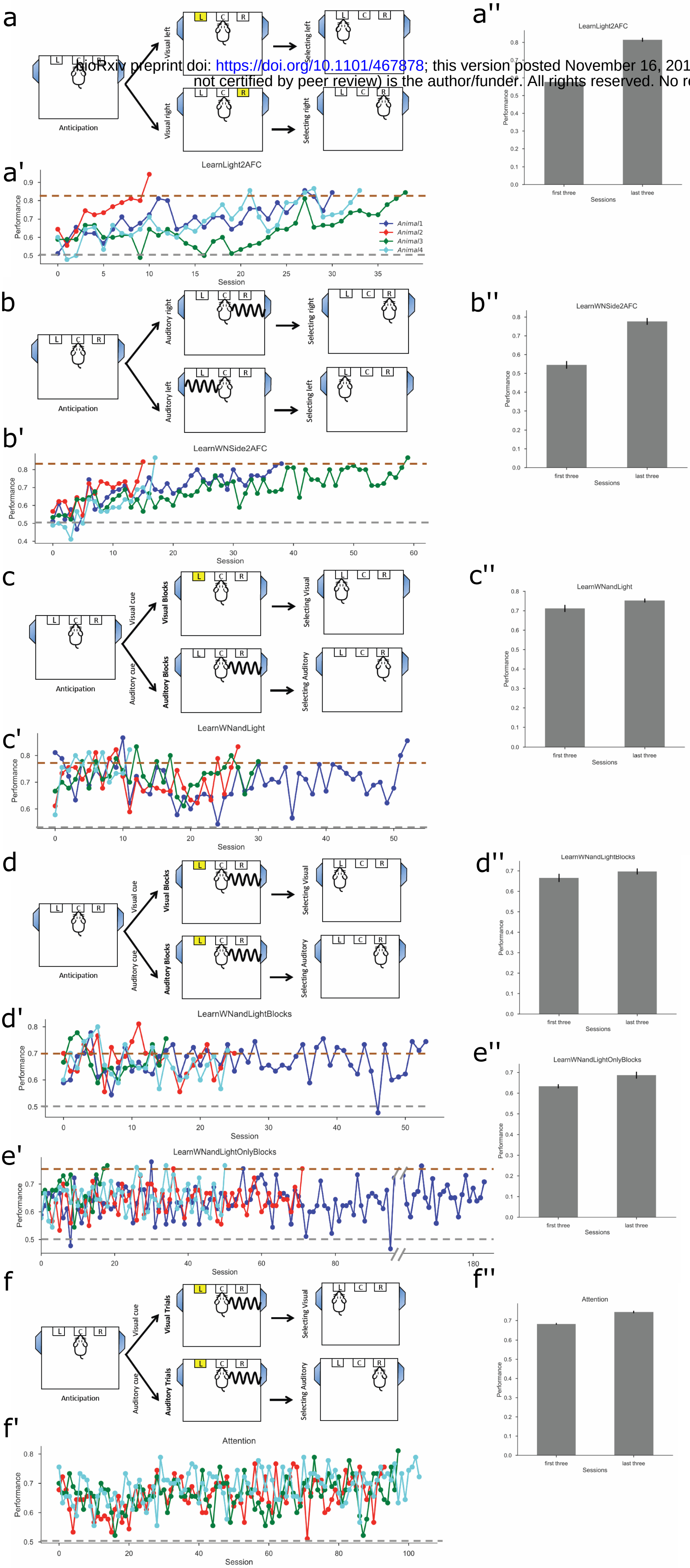
Supplementary Figure 1



Supplementary Fig. 1: Technical details of the hardware design

(a-b) Side view of the setup (**a**) packed into a light- and sound-proof box (**b**). **(c)** RFID sorting process. For an animal to enter the behavior box, only when the left and the middle detectors detect the same RFID chip, the left gate is closed and the right gate is open so that the animal can access the behavior box. For an animal to return to the home cage, only when the right and the middle detectors detect the same RFID chip, the right gate is closed and the left gate is open so that the animal can go back to the home cage. In the entry and the return processes, if the left and the middle detectors detect different RFID chips, the animals have to leave the tube and the detectors get reset afterwards. **(d)** Schematic of RFID access control circuit. **(e)** Schematic of the software controlling the devices. A master program receives input from the RFID sorting device and controls four other modules including Bpod, synchronized video recording, data management, and logging. **(f)** Top view of a Raspberry Pi version of the setup.

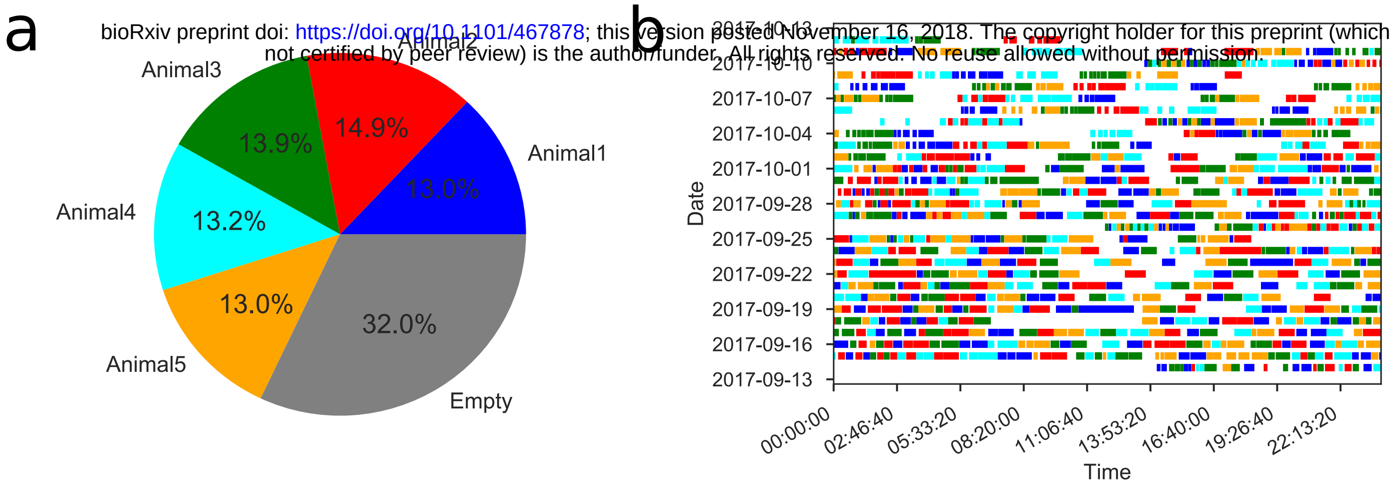
Supplementary Figure 2



Supplementary Fig. 2: Illustration of training procedures

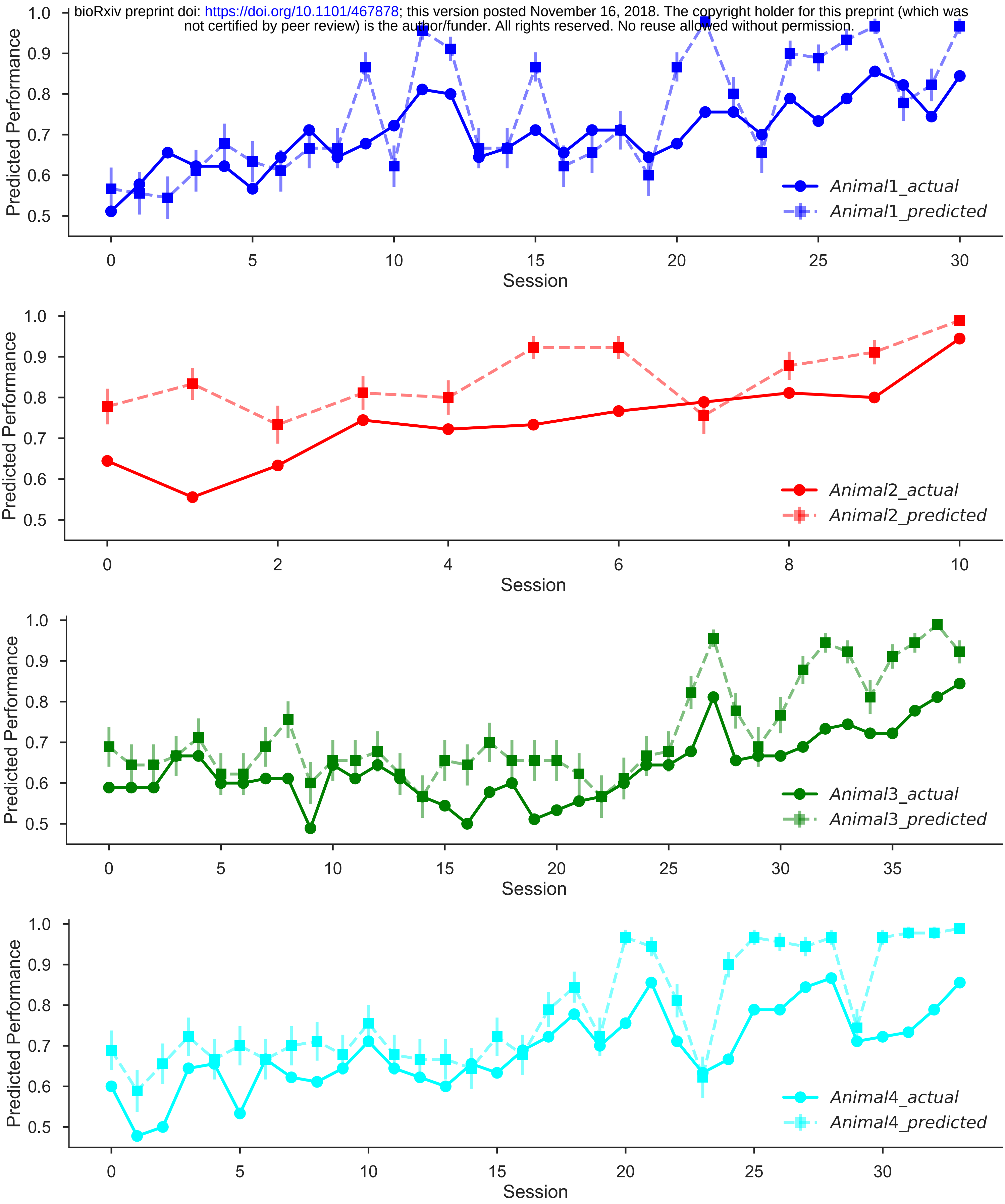
Training proceeds through six stages (Online methods). The design, learning curves, and animal performance of the simple visual task **(a, a', a'')**, the simple auditory task **(b, b', b'')**, the cued single-modality (visual or auditory) switching task **(c, c', c'')**, the cued single- (visual or auditory) and double-modality (attend to vision or audition) switching task **(d, d', d'')**, the cued double-modality (attend to vision or audition) switching task **(e', e'')**, and the final selective attention task **(f, f', f'')** are shown here. **a'** displays performance data as in **Fig. 3e**. Brown and gray dashed lines indicate the performance thresholds for upgrading to the next stage and downgrading to the previous stage respectively.

Supplementary Figure 3



Supplementary Fig. 3: Automated training system allows efficient use of the behavior box
For a sample cohort of five animals, this shows the fraction of time each animal used the behavior box (**a**) and the activity trace of each animal throughout one month.

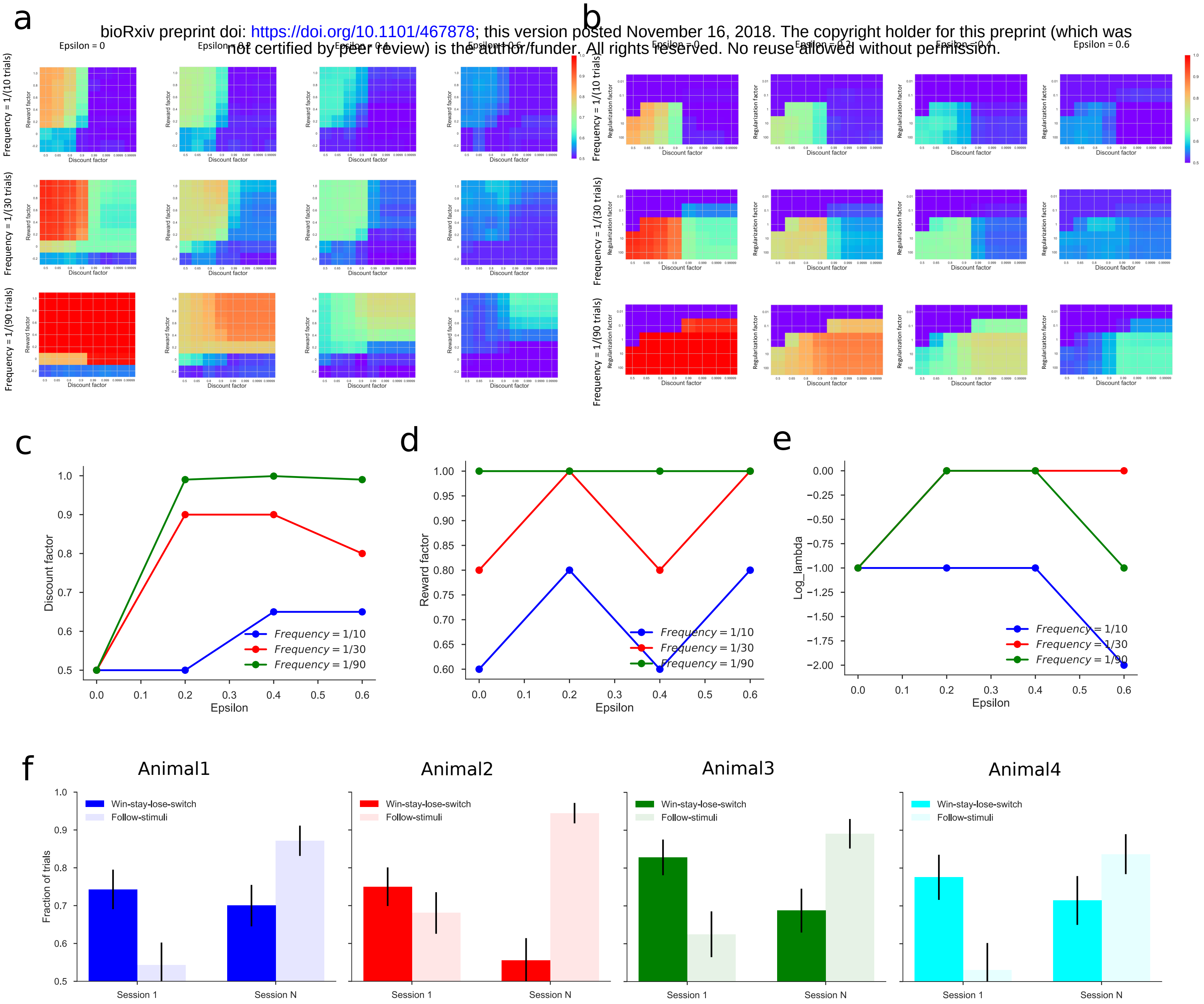
Supplementary Figure 4



Supplementary Fig. 4: Additional analysis on the iterative generalized linear model's prediction accuracy

The actual performance and the performance predicted by the model, for each of the four animals. Note that the predictions recapitulate the more prominent fluctuations in the actual learning curves. Error bars indicate standard errors.

Supplementary Figure 5

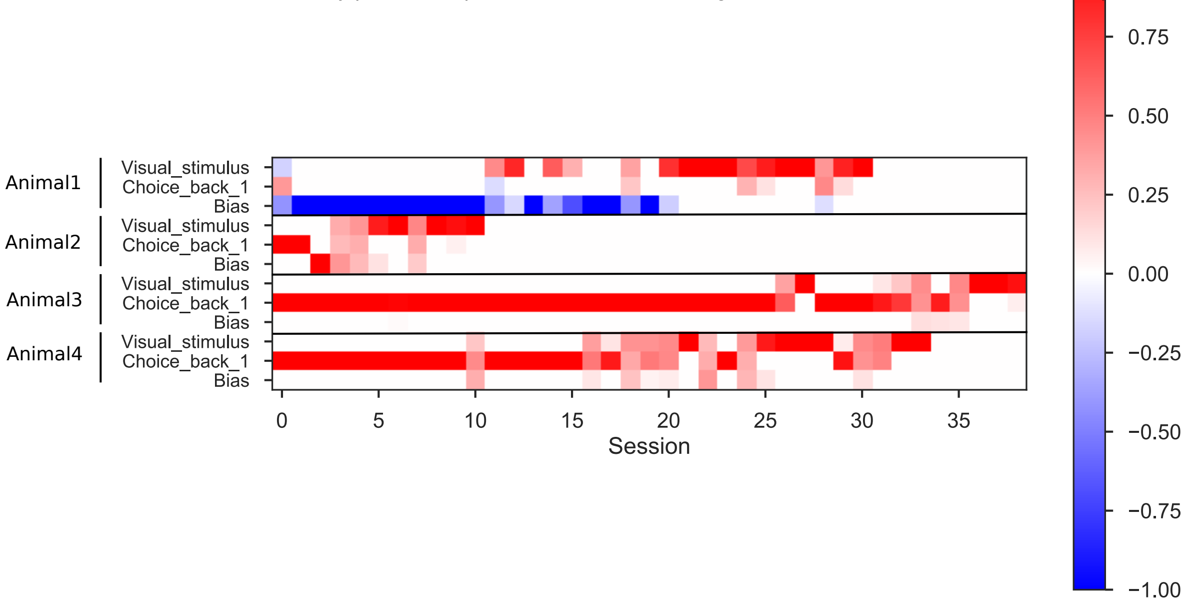


Supplementary Fig. 5: Iterative generalized linear model captures differences between individuals and policy changes

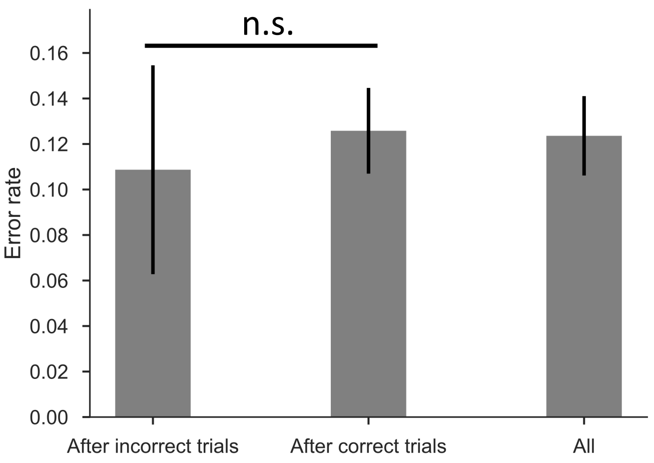
(a-b) Hyperparameter selection for the GLMs fitted to the simulated data generated from the ground truth policies. Different values of policy change frequency and noise level (epsilon) lead to different landscapes of the hyperparameters. **(c-e)** Selected temporal discount factor (**c**), reward factor (**d**) and regularization factor (**e**) for different values of policy change frequency and noise level (epsilon). **(f)** Fraction of the trials explained by the two policies (win-stay-lose-switch or WSLS, and following the stimuli) in the first and last sessions, for each of the four animals.

Supplementary Figure 6

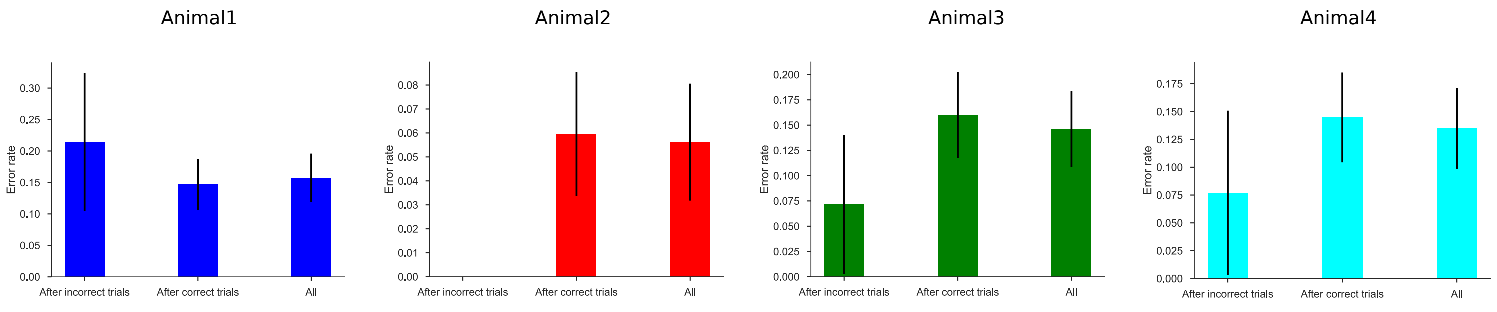
a



b



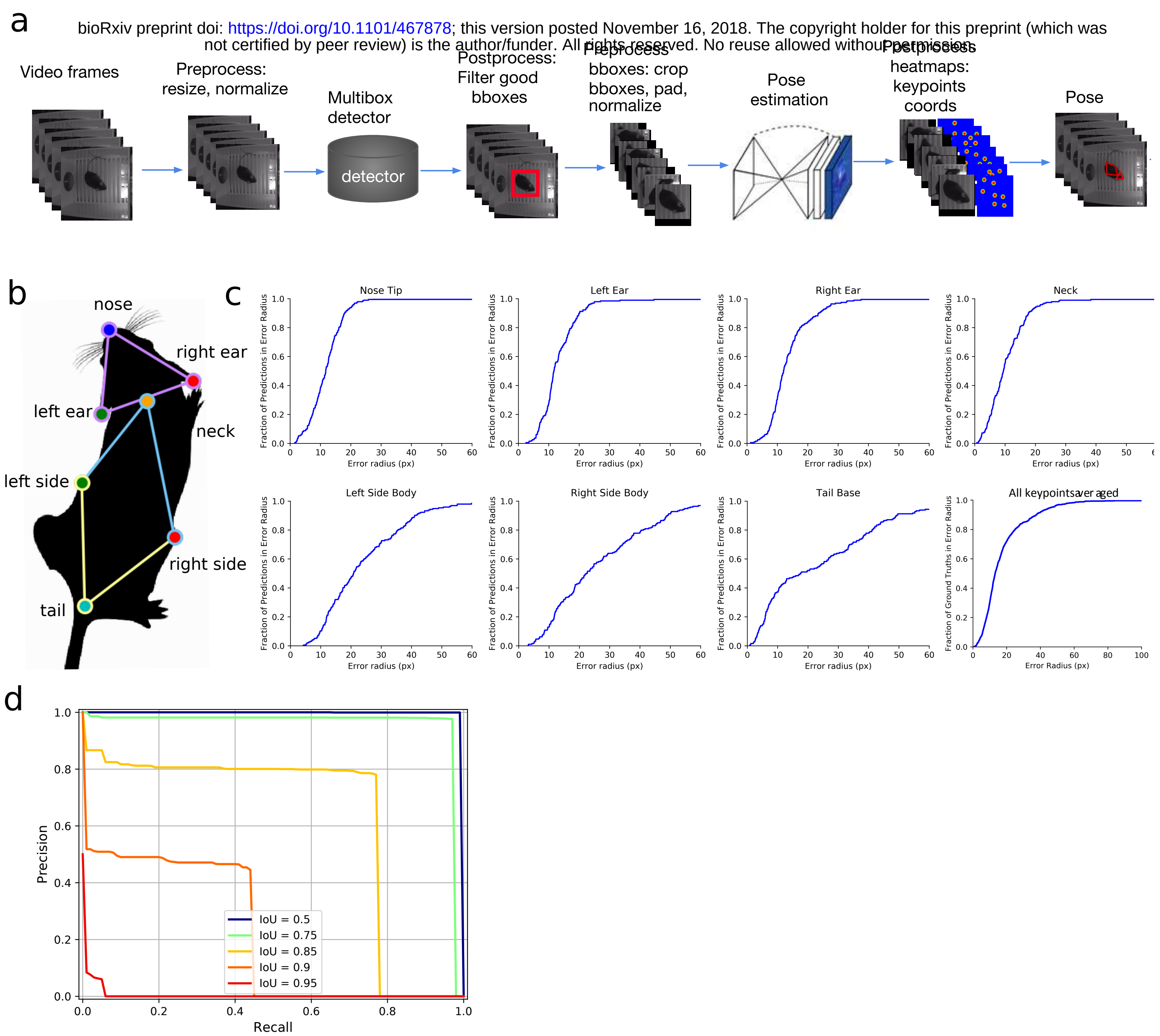
c



Supplementary Fig. 6: Additional analyses of policy changes during learning

(a) Policy matrices over sessions of the four animals. Here the policy matrices are recovered from logistic regression using only the trials following a correct response. Because the reward of the last trial is always +1, the term Reward_back_1 is the same as Bias, and the term RewardxChoice_back_1 is equal to Choice_back_1, so we drop them to avoid redundancy. **(b-c)** Quantification of the error rate during the last session, comparing trials following a correct response to those following a mistake. Averaged over all four animals (**b**) and for each of the four animals (**c**). n.s. indicates not significant.

Supplementary Figure 7

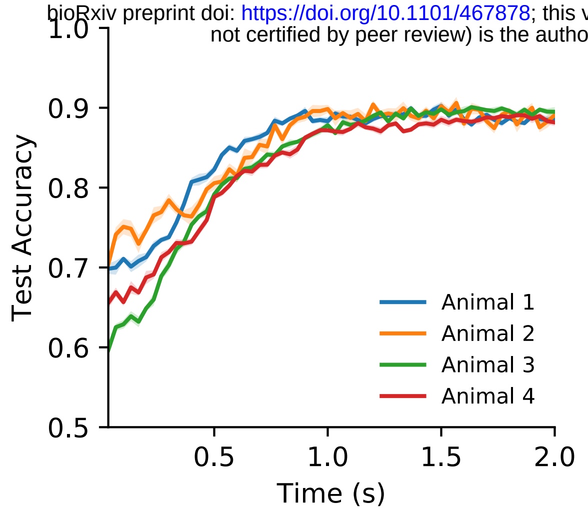


Supplementary Fig. 7: Technical details of the automated behavior assessment software

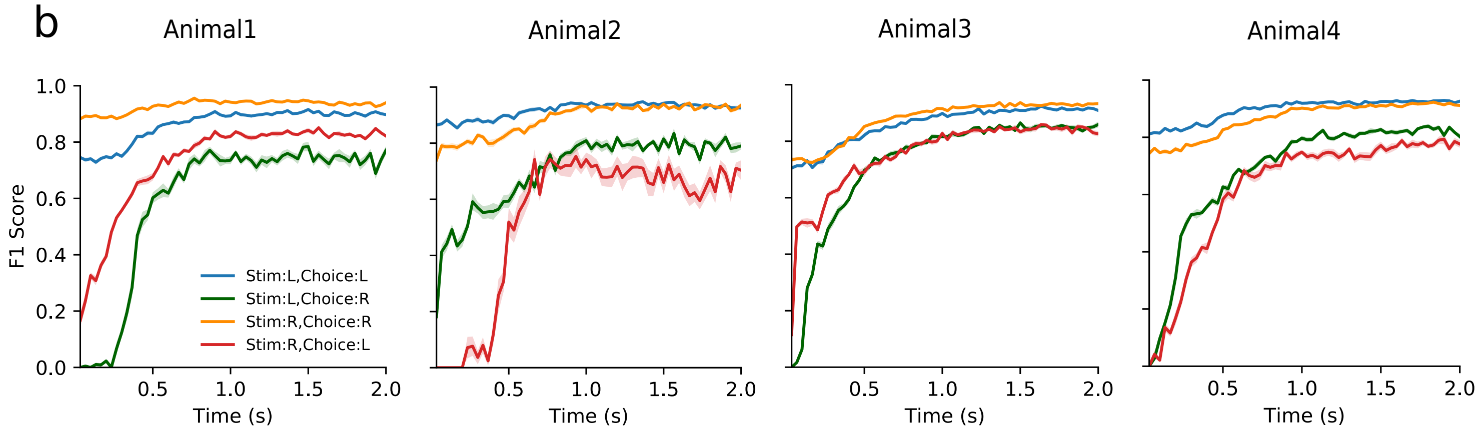
(a) Detailed illustration of the automated behavior assessment software as presented in **Fig. 5a**. Video frames are first preprocessed (resized and normalized) and then passed through the MSC-Multibox detector (the first deep neural network (DNN)) to generate bounding boxes covering the mouse body. Then images of the mouse around the bounding box are cropped, padded and normalized for pose estimation. Pose estimation is done by Stacked Hourglass Network (the second DNN), which produces seven keypoint coordinates for body landmarks as shown in **b**. **(b)** The seven body landmarks of the nose, the ears, the neck, the body sides, and the tail. **(c)** Percentage of Correct Keypoint (PCK) curves. The fraction of inferred keypoints that lie within a specified pixel radius of the true location (10 pixels = 0.25 cm). On average over all seven keypoints ~80% lie within an error radius of 0.5 cm. **(d)** Precision-Recall curves at different thresholds of Intersection over Union (IoU) w.r.t ground truth bounding box. An IoU of 75% already detects mice very effectively, with precision and recall above 95% .

Supplementary Figure 8

a



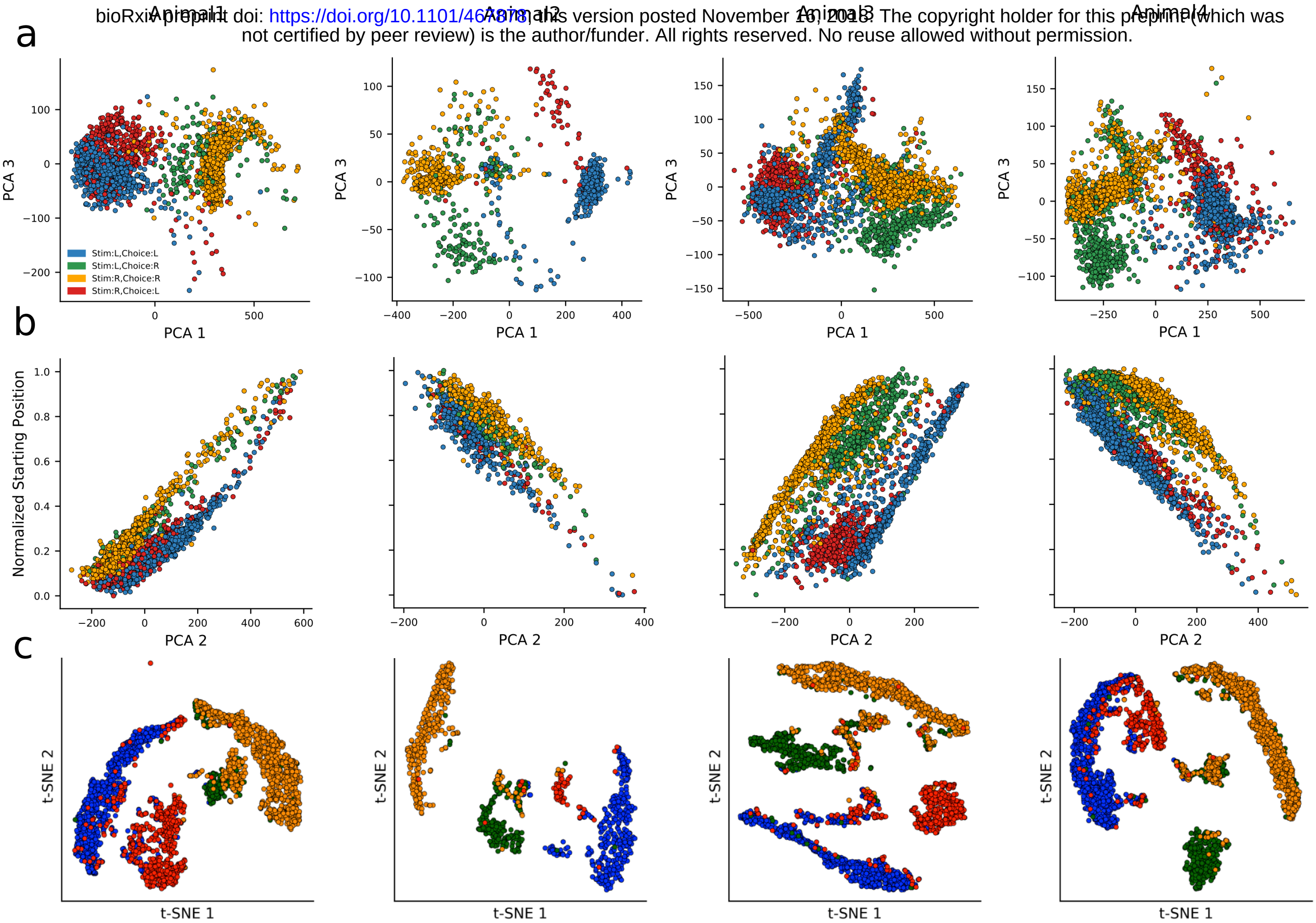
b



Supplementary Fig. 8: Performance of the support vector machine to infer trial category from mouse trajectories

(a) Prediction accuracy of the SVMs for individual animals. **(b)** F1 score of the SVM fitted for the decision categories of each animal. Shaded region denotes standard error.

Supplementary Figure 9



Supplementary Fig. 9: Additional information on the unsupervised analysis of behavior trajectories

(a) Projection of trajectories onto PC1 and PC3. Different decision categories are indicated by different colors, which is the same for **b** and **c**. **(b)** Scatter plot of starting positions along the left-right axis against PC2 shows correlation between the two. Starting positions are normalized to range from 0 (the leftmost position) and 1 (the rightmost position). **(c)** t-SNE plots with colors indicating different decision categories.

Supplementary Video 1:
https://drive.google.com/open?id=1Ng5s1UhlFRdV4mZ5b1Ot7EUpaEiUXN2_

Supplementary Video 2:
https://drive.google.com/open?id=15qgqM5qOd30kajT-IQkCf8flcF80U2qV

Supplementary Video 3:
https://drive.google.com/open?id=1zqja6_3bA2jO9ap0EWd5t_Z_dVOA8FmA

Supplementary Video 4:
https://drive.google.com/open?id=1wiaaBD-sfZTudDbpRM0ByB2n9bTKyaid

Supplementary Video 5:
https://drive.google.com/open?id=1yjaZXDUS3TKzDR4YGHYUlbwbq0OfUmI4

Supplementary Video 6:
https://drive.google.com/open?id=1ISELpiyI0Yh_EoiGn7aX1GLSoUaFtjKa

Supplementary Video 7:
https://drive.google.com/open?id=1dILuZRlwAU48RwLEEkt1CBzmPQnqF95r