Title
An open-source behavior controller for associative learning and memory (B-CALM).

Journal
Behavior Research Methods, 56(4)

Authors
Zhou, Mingkang
Wu, Brenda
Jeong, Huijeong
et al.

Publication Date
2024-04-01

DOI
10.3758/s13428-023-02182-6

Peer reviewed

HHS Public Access
Author manuscript
Behav Res Methods. Author manuscript; available in PMC 2025 April 01.

Published in final edited form as:
Behav Res Methods. 2024 April ; 56(4): 2695–2710. doi:10.3758/s13428-023-02182-6.

An open-source Behavior Controller for Associative Learning and Memory (B-CALM)

Mingkang Zhou[1,2,*], **Brenda Wu**[1,*], **Huijeong Jeong**[1], **Dennis A Burke**[1], **Vijay Mohan K Namboodiri**[1,2,3,#]

[1]Department of Neurology, University of California, San Francisco, CA, USA

[2]Neuroscience Graduate Program, University of California, San Francisco, CA, USA

[3]Weill Institute for Neuroscience, Kavli Institute for Fundamental Neuroscience, Center for Integrative Neuroscience, University of California, San Francisco, CA, USA

Abstract

Associative learning and memory, i.e., learning and remembering the associations between environmental stimuli, self-generated actions, and outcomes such as rewards or punishments, are critical for the well-being of animals. Hence, the neural mechanisms underlying these processes are extensively studied using behavioral tasks in laboratory animals. Traditionally, these tasks have been controlled using commercial hardware and software, which limits scalability and accessibility due to their cost. More recently, due to the revolution in microcontrollers or microcomputers, several general-purpose and open-source solutions have been advanced for controlling neuroscientific behavioral tasks. While these solutions have great strength due to their flexibility and general-purpose nature, for the same reasons, they suffer from some disadvantages including the need for considerable programming expertise, limited online visualization, or slower than optimal response latencies for any specific task. Here, to mitigate these concerns, we present an open-source behavior controller for associative learning and memory (B-CALM). B-CALM provides an integrated suite that can control a host of associative learning and memory behaviors. As proof of principle for its applicability, we show data from head-fixed mice learning Pavlovian conditioning, operant conditioning, discrimination learning, as well as a timing task and a choice task. These can be run directly from a user-friendly Graphical User Interface (GUI) written in MATLAB that controls many independently running Arduino Mega microcontrollers in parallel (one per behavior box). In sum, B-CALM will enable researchers to execute a wide variety of associative learning and memory tasks in a scalable, accurate, and user-friendly manner.

[#]Correspondence to VijayMohan.KNamboodiri@ucsf.edu.
[*]These authors contributed equally to this manuscript

Introduction

Many motivated behaviors require animals to learn and remember the associations between environmental cues, actions, and/or rewarding/punishing outcomes. Hence, the study of associative learning and memory is a cornerstone of psychology and behavioral neuroscience. An investigation of these behaviors requires a scalable, accurate, and cost-effective behavior control system (hardware and software) that can interface with brain recording and manipulation. These systems should have sub millisecond timing accuracy, be flexible enough to control multiple behaviors, and ideally be easy to set up for users without extensive technical and programming expertise. However, combining flexibility, scalability, affordability, and ease-of-use in a single behavioral control system remains challenging.

Historically, the most common solution to this problem has been to use proprietary commercial control systems. While these systems are accurate, modular, and easy to set up, they are expensive. The high cost of these systems limits their scalability to conduct many behavioral experiments in parallel and increases the barrier to adoption. Further, troubleshooting may require specialist support, which can be challenging during time-sensitive experiments.

To avoid using costly proprietary systems, many labs develop their own custom software and hardware setups. A common approach has been based on National Instruments data acquisition cards and software (Labview) (e.g., Elliott et al., 2007; Kim et al., 2015; Mohebi et al., 2019; Nikbakht and Diamond, 2021), or real-time operating systems (e.g., Brunton et al., 2013; Jaramillo and Zador, 2011; Poddar et al., 2013). However, these require considerable in-house technical expertise to develop and implement. With the advent of easily customizable microcontrollers (e.g., Arduino or Pyboard) (e.g. Belsey et al., 2020; D'Ausilio, 2012; Namboodiri et al., 2019; Schultz and van Vugt, 2016; Toda et al., 2017; Woods et al., 2020) or microcomputers (Raspberry Pi), multiple general-purpose and open-source behavioral control systems have recently been developed. A few examples include Bpod (SanWorks LLC) (Solari et al., 2018), Autopilot (Saunders and Wehr, 2019), and pyControl (Akam et al., 2021). These systems are highly flexible and powerful and provide nice solutions for expert programmers to develop their own custom tasks. Nevertheless, they have some limitations resulting from their general-purpose design. First, their general-purpose design, while providing flexibility, limits response latencies. Second, they are not set up to directly run the gamut of associative learning and memory tasks, thereby requiring considerable programming experience and time commitment to develop a fully functional controller for multiple associative learning and memory tasks. Third, they either do not provide quick visual summaries of behavioral performance during data acquisition or require considerable additional programming to this end.

To solve these problems, we provide an open-source behavioral control system using a low-cost Arduino Mega 2560 board interfacing with a MATLAB Graphical User Interface (GUI). This system allows a single standard computer to control multiple behavior boxes in parallel, with each box running an independent experiment (one Arduino Mega 2560 per behavior box). Our system contains three main components (Figure 1): (1) the MATLAB GUI, (2) Arduino Mega 2560 and electronic circuit setup, and (3) the behavioral box setup.

User interactions are done solely through the GUI to minimize operational complexity. Task commands, such as file loading and parameter setup, are inputted through the GUI (Figure 2) and subsequently transmitted to the Arduino Mega 2560 via a USB serial connection. As the Arduino Mega 2560 executes the task based on the appropriately loaded control program, it conveys information to the computer and these task details are plotted in real time on the GUI (Figure 2).

We first show that the timing accuracy and response latency of the system is much smaller than a millisecond—the typical fastest timescale of neuronal activity change. To illustrate the functionality of the system, we trained head-fixed mice to perform five standard associative learning tasks including Pavlovian conditioning, operant conditioning, discrimination learning, peak interval timing, and a two-option choice task. Each of these tasks is introduced in detail below.

Pavlovian conditioning: Environmental cues and contexts can influence our behavior when they become predictive of future outcomes such as rewards or punishments. The learning of such cue-outcome associations is fundamental for successful reward seeking. These processes can also result in maladaptive memories in neuropsychiatric disorders such as addiction or post-traumatic stress disorder. For example, individuals with substance use disorders may treat physical items such as needles and less tangible sensory features of previous drug usage locations as drug-associated stimuli, prompting behaviors such as craving and relapse (Goldstein and Volkow, 2002; MacNiven et al., 2018; Perry et al., 2014). Despite the ubiquity of such learning and memory, the brain mechanisms that contribute to them remain to be fully worked out. Hence, Pavlovian conditioning tasks are widely used to study associative learning and memory.

Operant conditioning: Also known as instrumental learning, operant conditioning is a form of learning in which animals increase or decrease the rate of performance of actions if an appetitive or aversive stimulus is contingent on that action. If the outcome is the receipt of a reward, or the avoidance of a punishment, the action is reinforced, whereas if the outcome is the avoidance of a reward, or the receipt of a punishment, the action is progressively suppressed. Common operant conditioning paradigms in rodents train animals using lever pressing or entries to a nosepoke in freely moving preparations (e.g., Matikainen-Ankney et al., 2021). In head-fixed mice, animals have been trained to perform actions such as licking, lever pressing, joystick manipulation, or turning a wheel (Burgess et al., 2017; Gordon-Fennell et al., 2023; Livneh et al., 2017; The International Brain Laboratory et al., 2021; Vollmer et al., 2020; Yttri and Dudman, 2016).

Discrimination learning: In real life, actions often result in outcomes only if performed after specific environmental cues. Such behavior requires discrimination learning. The cue in such cases is often called a discriminative stimulus (DS) as it signals that performing a specific action immediately following the cue can result in an outcome. In principle, animals can be trained on this task by initially learning an operant response, followed by learning to produce the operant response to a specific cue, or by transitioning from learning a Pavlovian cue-outcome association to discriminative learning (Crouse et al., 2020; Ghazizadeh et al., 2012; K Namboodiri and Stuber, 2021; Richard et al., 2016).

Peak interval timing: Studies investigating reward-based behavior do not often consider the role of time perception, even though time perception is fundamental to behavior production in every task we have discussed so far. A common behavioral task to study timing behavior is the peak-interval timing task, which is a variation of a fixed interval operant task with "peak" reward omission trials that test if the animal learns the expected delay to rewards (Matell and Portugal, 2007; Roberts, 1981). This has recently been modified to a Pavlovian version in which head-fixed mice learn a fixed delay between rewards and their timing behavior is measured on some probe trials without rewards (known as peak trials) (Toda et al., 2017).

Two-option choice task: Though each of the above tasks has a component of decision-making (when and whether to perform an action), they do not directly evaluate decisions between alternative actions. A common approach to this end is to study two-option choice tasks in which animals choose between two possible outcomes (Herrnstein, 1961). For instance, animals can choose between different rewards of varying reward magnitude or quality (e.g., apple juice versus orange juice, or alcohol versus sucrose), probability (often referred to as a two-armed bandit task), or delays (intertemporal choice tasks), or combinations of these variables in tasks such as a delay discounting task (Ainslie and Herrnstein, 1981; Ainslie, 1974; Chung and Herrnstein, 1967).

We provide the code and documentation to implement these and other tasks (Table 1), and all supporting information for hardware and software setup (https://github.com/namboodirilab/B-CALM). In combination, these tools will enable researchers to execute a wide variety of associative learning and memory tasks in a scalable, accurate and user-friendly manner. Thus, we believe that our system provides a valuable addition to the space of open-source behavioral control systems.

Methods:

Data and Code availability:

All supporting information is available on the Namboodiri lab GitHub page (https://github.com/namboodirilab/B-CALM).

Experimental subjects and surgery

The behavioral tasks described in this paper used adult wild type C57BL6/J mice from Jackson Laboratories. Mice were 8-9.5 weeks of age at the time of surgery and 10-12 weeks at the commencement of behavioral experiments. Specifically, we ran 8 male and 3 female mice on the Pavlovian conditioning task, 3 male and 4 female mice on the sequential conditioning task, 5 female mice on the operant conditioning task, 6 mice on the discrimination learning task (n=2 from Pavlovian group, n=4 from operant group), 4 female mice on the choice task, and 5 male mice on the interval timing task. All experimental procedures were approved by the Institutional Animal Care and Use Committee at UCSF. 8-week-old mice were anesthetized with isoflurane (~1.5%) and then held in a stereotaxic apparatus (David Kopf Instruments). Mice were kept on a heating pad to maintain warmth, ophthalmic ointment was applied to protect the eyes, and pedal reflex was checked prior to surgery. Analgesia was used upon the start of the surgery and also during post-operative care (lidocaine 1mg/kg, carprofen 5 mg/kg, and buprenorphine 0.1 mg/kg). A custom-made stainless-steel head-fixation ring was implanted on the skull using dental cement. Animals were carefully monitored for 3 days post-surgery.

Behavioral apparatus

Arduino Mega 2560 is a single board microcontroller equipped with 54 digital inputs/output pin and 16 analog input pins and can interface with other electronic devices. Our control system utilizes an Arduino Mega 2560 board in each behavioral box. Here, we have a single standard computer controlling multiple Arduinos at once. Each Arduino Mega 2560 board is connected to the computer via separate USB cables, meaning that multiple behavior boxes can be controlled in parallel.

In this system, Arduino digital pins are directly connected to a breadboard using wrapping wires, which helps build all the essential circuits needed for hardware such as lick spout detectors, speakers, LED lights, solenoids, lasers, and more. The wires are connected to a breakout terminal block that is glued onto the outside of a plastic box holding the Arduino and breadboard. Pins in the breadboard (BB1660T, BusBoard Prototype Systems) can then be connected to their corresponding slots on the interior portion of the breakout terminal. From here, all hardware can be connected to the appropriate terminal slots on the exterior portion of the box. All detailed instructions with photo illustrations are provided on the lab GitHub page in the “circuit schematic.pdf” document (https://github.com/namboodirilab/B-CALM/blob/main/circuit%20schematic.pdf).

We decided to build long-term stable designs using breadboards, stable breakout pins and wire wrapping, instead of printed circuit boards. This is primarily to allow flexibility of circuit modifications, while retaining stability (see the circuit schematic file on the lab GitHub page for more details). If a hardware is no longer functional, the user can easily replace it with a new hardware. Furthermore, depending on a user’s research interests, the design of a behavioral task can be modified by simply connecting additional input hardware. As such, only external connections need to be changed—everything inside the box remains the same. This helps to ensure that all wire connections inside the box stay intact and stable, which allows for long-term use. Since digital signal changes in a behavior controller only occurs in the millisecond range, stray capacitances on a breadboard are not of much concern. Lastly, the breadboard listed above has high quality clips and is highly reliable in the long term.

To control the behavioral tasks, we set up our hardware to include two separate piezo buzzers/speakers (with different resonant frequencies: 2.56kHz/8kHz) (1739/1740, Adafruit), three lick tubes (left, right, center; center lick tube can deliver two different liquids if needed) with three lick detector circuits (licks on the left lick tube are registered as “lick1s”; licks on the right lick tube are registered as “lick2s”; licks on the center lick tube are registered as “lick3s”) (Slotnick, 2009), one fluid delivery solenoid (003-0257-900, Parker Hannifin) for the center lick tube (our default center solenoid is set as solenoid3), two extension/retraction solenoids for the side lick tubes (TAU0730TM-14,Chaocheng Technology), and two LEDs (PWDIODE-30-Blue, ToToT). All detailed instructions for setting up hardware in the GUI parameters, full parameters explanation, and task running overview is provided on the lab GitHub page in the “Task running overview.pdf” document (https://github.com/namboodirilab/B-CALM/blob/main/Task%20running%20overview.pdf). All equipment was set up on a custom-designed head-fixation platform. The position of all lick tubes was controlled precisely in three dimensions, using combinations of Thorlabs positioning systems (MS1S, Thorlabs). Based on our experience, such precise placement of lick tubes is essential for repeatable and stable behavior in these tasks (see https://github.com/namboodirilab/B-CALM/blob/main/head_fixed_setup/Head-fixed%20stage%20setup%20overview.pdf for more information on lick tube placement). All apparatus was placed in a sound proofing box (1430-MGY, Tennsco) and covered with sound proofing foam (B08HCTMPDX, Acoustic Foam). A detailed parts list and set of instructions for building the hardware setup is provided on the lab GitHub page in the “Head-fixed stage setup overview.pdf” document (https://github.com/namboodirilab/B-CALM/blob/main/head_fixed_setup/Head-fixed%20stage%20setup%20overview.pdf).

Graphical User Interface

User interactions are done solely through the MATLAB GUI to minimize operational complexity and optimize experimental flexibility. This user-friendly GUI allows the set-up of behavior parameters and visualization of behavioral performance to be performed in one convenient setting (Figure 2). We decided to use MATLAB instead of an open-source language such as Python to allow for backward compatibility and long-term stability of the code without continuous codebase maintenance.

To activate the GUI, the user must first select a vacant Arduino board. Here, we refer to this as “COMX,” where COM stands for communication port and X corresponds to the board being used. The next steps include choosing the desired behavioral experiment mode, uploading the respective Arduino file to that board, and linking the serial connection between the computer and Arduino board (Figure 2A). This allows commands from MATLAB to be sent to the Arduino. Detailed instructions to upload the correct Arduino file based on the experiment mode is provided on the lab GitHub page in the “Task running overview.pdf” document (https://github.com/namboodirilab/B-CALM/blob/main/Task%20running%20overview.pdf).

The corresponding areas in the GUI used in defining task parameters will activate and allow for variable specification (Figure 2B). For instance, a random rewards task (e.g., 15% sucrose reward delivered based on a truncated Poisson process with a 12 s mean interval) will use the section titled “Background rewards.” These rewards are labeled background rewards because they can be delivered during the background inter-trial interval of trial-based experiments. The amount of reward given is based on the background solenoid’s open time, a parameter that the user must define prior to the start of the task. All time values used are in units of milliseconds.

Next, the user can send all task parameters to the Arduino by clicking the “Send” button (Figure 2F). After sending the parameters, the user can ensure hardware functionality by testing them before initiating the experiment (Figure 2C). For example, pressing on “Test CS1” will turn on the CS1 cue for one second (this set length of time is not modifiable through the GUI but can be changed in the Arduino task scripts if needed). Moreover, pressing on “Manual” under Solenoid3, which is the default solenoid connected to the sucrose reward delivery system, allows the user to prime animal subjects with sucrose rewards before starting the behavioral session. The “Start” button launches the task (Figure 2F), and then the Arduino will carry out the operation and activate hardware accordingly based on the task design.

As the behavioral task is in progress, the Arduino records all events detected by the hardware system and sends them back with timestamps to the GUI. Live events (e.g., number of auditory or light cues presented, licks performed, solenoid openings, etc.) are counted and displayed in real time on the GUI, along with its corresponding plot legends (Figure 2D, E, G). The real-time plotting can be observed for each behavioral box simultaneously across independent GUIs controlling each box.

Once a behavioral session is over, pressing “Unlink” (Figure 2A) will unlink the serial connection between the computer and Arduino board. As such, the user will then be able to select a different Arduino board (if needed) or continue with the same board, choose either the same experiment mode or another one, upload the same Arduino file or another Arduino file, and link the serial connection once again to initiate the next session.

A MATLAB file is automatically generated at the end of a session (even when there are errors). The filename contains the subject’s name, experiment mode, and date/time information. When the session ends, the data file saves and stores two variables: (1) params; which contains all task parameters for the session, and (2) eventlogs; which records all event identifiers, timestamps, and an additional flag (0 or 1) for each event. This item flag indicates 1) reward delivery status of all solenoid events: 0 means reward delivery; 1 means no actual reward is delivered, or, 2) cue order index for trials with two cues: 0 indicates the first cue, 1 indicates the second cue. All the information necessary for data analysis is accessible through these two variables to simplify data handling.

Training procedure

Animals were water deprived prior to the commencement of behavioral training. We targeted to maintain body weights at a stable level of 85-88% of their pre-deprivation weight. To this end, animals were provided supplemental water in addition to the sucrose earned during behavioral sessions. The amount of supplemental water given to each animal was adjusted daily to maintain their weights around 85-88% of pre-deprivation weight. Prior to both the Pavlovian and operant conditioning tasks, animals were first trained to lick on the lick tube delivering sucrose. This is achieved by delivering random Poisson trains of 15% sucrose drops (measuring 3 μL, this is calibrated combining the height of the sucrose syringe and the solenoid open time) with a mean inter-reward interval of 12 s for a total of 100 rewards (i.e., approximately 20-minute sessions) for at least two consecutive days. After the animals successfully learned to lick on the lick tube by performing more than 1000 licks per session, they were considered ready to progress to the actual behavioral tasks.

For Pavlovian conditioning, mice were trained to lick in response to an auditory conditioned stimulus predictive of sucrose reward (CS+), but not to another auditory stimulus that predicted the absence of sucrose reward (CS−). CS+ and CS− tones were either a 12 kHz continuous tone or a 3kHz pulsed tone. To control for cue identity bias, we assigned the 12 kHz stimulus as CS+ for half of the mice and the 3 kHz stimulus as CS+ for the other half. Each session included a total of 100 trials (50 CS+, 50 CS−), all of which were presented in pseudo-random order. A trace interval of 1 s followed the offset of the auditory cues (2 s) until either sucrose reward delivery or omission. There was an exponentially distributed intertrial interval of 30 s in addition to a fixed reward consumption time of 3 s. Each session lasted approximately one hour. Each animal ran for one session per day, with occasional breaks on weekends. For the Pavlovian conditioning data shown in Figure 4D, five animals on average took 11.8 days to be fully trained, at which point they licked in anticipation of sucrose for CS+ cues only. Six animals were separately run for ten days with 100 CS+ trials per day (no CS− trials) (Figure 4E). For sequential conditioning, we followed previously described methods (Jeong et al., 2022). These data are the same as those shown for the control animals in Figure 6 of Jeong et al., 2022. Please note that the training protocol described here has been improved upon recently (Burke et al. 2023).

For the operant task, we trained side licks as the operant action. Importantly, to ensure that the operant action is separable from a consummatory action, we trained animals to lick on a side lick tube (which never delivers any outcome) to obtain rewards on a center lick tube (Figure 4G). Our setup allows the conditioning to be run using fixed ratio (side lick a fixed number of times to get reward from center tube), random ratio (side lick results in reward at a given probability), fixed interval (first side lick after an interval expires from previous reward results in reward), random interval (first side lick after an unpredictable interval expires from previous reward results in reward), or combinations of these schedules. Progressive ratio schedule is also possible. For the data shown here, head-fixed animals were first trained on a fixed ratio (FR) 1 task, which they learned in a few minutes. Next, the fixed ratio criterion was systematically extended to FR 5. The average training length across five animals was approximately six days. Lastly, we performed a single random ratio 5 task to check whether mice performed operant licking despite uncertainty in the reward contingency.

As a proof of principle that B-CALM can control discrimination learning experiments, we trained two separate groups of mice following prior training on either Pavlovian conditioning or operant conditioning (Figure 5A, B). After the two groups learned their initial task, all mice were given the complete discrimination learning task. In this task, mice needed to perform the side operant lick after an auditory cue to receive reward in the middle lick tube. Since the operant group animals first learned to perform the side lick to acquire rewards, they had to then learn to perform this licking action after an auditory cue to obtain reward in the full task. Contrastingly, the Pavlovian group was initially trained to associate an auditory cue with reward and then were subsequently trained that these cue presentations must be followed by the operant lick to receive reward. To ensure that they knew the action of side licking, we performed one to two sessions of operant conditioning on these mice after they fully learned the Pavlovian association.

The task itself consisted of three stages: (1) Initial respective tasks: Pavlovian or operant. (2) Side lick tube extension on the discriminative stimulus indicating reward availability (DS+

cue), after which mice had to lick once to obtain reward. Tube extension trainings were done using an incremental scale, by placing the extended lick tube zero, half or more than a full tongue width (~4-5 mm) away from the center lick tube. The side tube extension initially covered the middle lick tube, so mice will lick both tubes simultaneously when licking the center lick tube. In the following days, we pulled back the side tube such that tube extension is eventually at least one tongue width away from the center (i.e., mice had to lick sideways to make contact on the side lick tube). (3) Full task; side lick tube extended on both DS+ and DS− cues at the furthest extension. Reward was given only if the FR lick requirement was met in response to DS+. No reward was given after DS− cues regardless of action. On average, the Pavlovian group took approximately 31 days and the operant group animals took 23.5 days to be fully trained on the task.

For interval timing task, we started our mice with a 10 s fixed-time-schedule at 100% probability, where 15% sucrose solution (3 μL) was delivered every 10 s for 150 trials. On average, training took ten days (Figure 6A). After animals showed high anticipatory licking in this condition, we followed with a 10 s fixed-time-schedule at 80% probability (reward omitted on 20% of trials; also known as peak trials) for approximately eight days. This was done to measure the animals' internal timing of when reward would be delivered. When animals demonstrated comparable levels of anticipatory licking for this condition on both rewarded and unrewarded trials, we then proceeded to test three fixed-time-schedules of 7.5 s at 80% probability, 15 s at 80% probability, and 5 s at 80% probability. On average, five animals took 13 days total to perform at comparable levels of licking on both rewarded and unrewarded trials during the above three fixed-time-schedules.

In rodent models of two-option choice tasks, it is important to ensure that animals do not have an intrinsic bias towards one or the other action regardless of their outcomes. Hence, here we validated a proof of concept for a two-option choice task (Figure 7A). In our task, mice received a fixed amount of reward at 1 s delay after licking a side lick tube and received varying amounts of reward at 1 s delay after licking the opposite side lick tube (Figure 7B). Rewards were always delivered from the center lick tube. In the first phase of the choice task, mice underwent training to lick to either side of the lick tube, using a variable ratio 5 schedule for two days each. This training was necessary to ensure that mice learned the association between licking on either side of lick tubes and receiving rewards. Once mice learned successfully to lick on both sides, they were trained on a fixed left trial session and a fixed right trial session on alternating days for a total of four days (two fixed left days and two fixed right days). On the fixed left trials, reward given after licking on the left lick tube was always 2.5 μL (30 ms solenoid open time), and reward given after licking on the right lick tube varied between 0 to 5 μL (0 to 60 ms solenoid open time). The opposite was true for the fixed right trials, so licking on the right lick tube always resulted in 2.5 μL of reward and licking on the left lick tube resulted in reward ranging from 0 to 5 μL. This allowed us to gauge whether mice will choose the side that delivers the larger reward regardless of whether it is left or right, and fixed or variable. Because it took mice a few trials to realize that a new condition was being presented during a session (e.g., switch from 10 ms open time to 50 ms open time), we established a minimum of 30 rewards total for each condition/block on the variable side. During each training session (one per day), there were approximately 210 trials. We analyzed choice data based on the last ten trials of a block. Since we validated this task to show no side preference in mice and clear sensitivity to reward magnitude, the task can be modified to add different probability, delay, or identities of rewards in the future.

Data analysis

All data were analyzed using custom written MATLAB scripts based on the log of individual events and their timestamps. All experimental parameters were extracted from the saved MATLAB file for each session. These timestamps were used to plot peri-stimulus time histograms (PSTHs) of licking behavior. For testing differences in lick count, we used one-tailed or two-tailed paired t test, depending on the hypothesis.

Results

Response Latency, timing accuracy, and parallel execution:

An important consideration for a successful behavioral control system is the input-output latency (e.g., the delay in delivering a reward after an action that should immediately result in a reward; see Methods for the electronic circuit and behavioral box setup). To quantitatively measure the input-output response latency of our system, we first tested the latency of producing an event (turn on solenoid) after the detection of an external event (licking action) in our setup. To detect the input and output events, we used an oscilloscope (SDS1202X-E, Siglent Technologies) and measured the voltage pulse generated from a mouse lick and the solenoid onset for a reward triggered immediately after the lick (Figure 3A). This was measured to be approximately 5 μs. Next, to measure timing accuracy, we delivered the reward at a fixed delay after lick detection. This delay was set to either 100, 500 or 1000 μs. In all cases, the mean timing error was less than 20 μs (Figure 3B). These measurements are consistent with the high timing accuracy of Arduino boards (D’Ausilio, 2012).

Next, we tested the signal transmission reliability and timing accuracy to validate our system’s ability to run multiple Arduino Mega 2560 setups simultaneously using a single computer. Here, six Arduino boards connected to the same computer separately passed 500 pings from 0s to 1500s at a period of three seconds. We found that all Arduino boards successfully communicated each of the 500 pings and the percentage of data drops was 0% (Figure 3C). Next, we checked the difference in recorded timestamps in MATLAB between each consecutive ping. We found this number to be an average of zero but with a spread less than 20 μs. Overall, these results demonstrate that our behavioral setup is reliable and accurate in its timing and data communication.

For neuroscience related behavioral control, it is essential to have the behavior control be synchronized with external data acquisition hardware. To demonstrate this, we tested the system’s ability to interface with an external fiber photometry system. We approached this by sending TTL pulses of aperiodic behavioral events (here, random reward deliveries) from the Arduino to the external acquisition system and record the timestamps of the same events separately using the clocks of Arduino Mega board and the external acquisition system (Figure 3D). We then aligned those timestamps in post-processing by performing a linear transformation of the timestamps of the external acquisition system to match the Arduino clock (Figure 3E). This approach is generalizable to synchronization with other systems.

Application examples

In the following sections, we describe and demonstrate how our system successfully runs a variety of commonly used behavioral paradigms for studying associative learning and memory. The first task we ran was Pavlovian conditioning (Fanselow and Wassum, 2016; Pavlov, 1927). Specifically, we ran trace conditioning with a one second trace interval (delay conditioning requires a simple change in task parameters). The second task was operant conditioning (Staddon and Cerutti, 2003; Thorndike, 1898). Currently, our setup can run fixed ratio, random ratio, fixed interval, random interval, or mixtures of these schedules. It can also run progressive ratio schedules. The third task was one in which animals performed discrimination learning. In this task, the animals had to perform an action after sensing a cue to obtain a reward. The fourth task we studied was a recently proposed variation of a peak interval timing task, which allowed us to measure time perception (Toda et al., 2017). Lastly, we also show data from a two-option choice task in which mice chose one of two actions with the more preferred outcome. Since our setup can deliver multiple different outcomes, it can be used to control task variants involving different types of rewards or a reward and punishment (quinine) (K Namboodiri et al., 2021), or orally consumable drugs such as alcohol (Laguesse et al., 2017) or fentanyl (Monroe and Radke, 2021).

Though our software and hardware interface can be modified for head-fixed or freely moving experiments, we chose to demonstrate its utility using head-fixed behaviors since these behaviors can be combined with advanced imaging approaches such as multiphoton calcium imaging (Namboodiri et al., 2019; Otis et al., 2017; Woods et al., 2020). Further, head-fixed designs allow better control of the sensory environment of animals, thereby reducing behavioral variability for these associative learning and memory tasks. In the sections below, we present results from the five types of tasks discussed above.

Pavlovian conditioning:

Here, we show that B-CALM can control Pavlovian conditioning experiments in head-fixed mice (Figure 4A). Animals successfully learned the task and showed high anticipatory licking between the CS+-reward interval (Figure 4A–E). Our setup allows us to vary the delay to rewards, cue duration, probability of reward, present multiple cue-outcome associations, deliver a punishment (quinine) instead of reward, reduce cue-reward contingency by delivering unpredictable rewards during the intertrial interval, or have two cues within a single trial (e.g., second order conditioning, blocking). For instance, animals acquired sequential Pavlovian conditioning (Figure 4F) (reproduced from control animals shown in Figure 6 of Jeong et al., 2022). Further, by systematically varying the ITI, we found that behavioral learning exhibits temporal scaling with respect to the ITI, such that the overall learning rate can be faster than ten trials (Burke et al., 2023).

Operant conditioning:

As proof of principle, we collected data from mice performing fixed ratio (FR5) and random ratio (RR5) schedules (Figure 4G, H). For the FR5 schedule, all four animals showed an increase in overall lick rate, demonstrating learning of the FR5 schedule. For the RR5 schedule, overall lick rate reduced but remained positive (Figure 4H).

Discrimination learning:

Both groups eventually performed the discrimination learning task appropriately, licking at a high rate on the side lick tube (operant licking) only following DS+, but not DS− (Figure 5C, D).

Interval timing task:

We implemented a previously established modification of a peak interval timing task in B-CALM (Figure 6A) and tested whether mice learn 5 s, 7.5 s, 10 s, and 15 s intervals. We found that they distinguished all intervals in their licking patterns on peak trials (Figure 6B, C).

Choice task:

We tested the reward magnitude on the variable side at which mice chose both sides equally. To control for side preference, we performed multiple repeats with each side lick tube being chosen as the fixed reward side. In both cases, we found that mice chose both sides equally when the reward magnitudes were equal (Figure 7C). Given the success of this task, our system can be used to test more choice conditions with varying reward magnitude, type, delay and/or probability by modifying the available parameters.

Discussion

The neural basis for associative learning and memory is a topic of great interest to many neuroscience and psychology laboratories. Hence, there is need for a user-friendly, affordable, flexible, and scalable system to control common behavioral tasks involving associative learning and memory. These tasks are routinely performed using proprietary hardware and software. However, these systems are expensive, costing tens of thousands of dollars. More recently, a lot of open-source behavioral control systems have been developed (e.g., BPod, PyControl, and Autopilot). When self-assembled, the cost of each of these systems is close to an order of magnitude lower than proprietary systems. For the specific electronic hardware configuration that we chose in our own studies, the cost is even lower than the above open-source control systems. A single BCALM controller system that produced the results described above had a total hardware cost including the Arduino/circuit board setup and electronic hardware of $303 (Table 2). The details of the circuit building and cost information for other parts is provided in the "circuit schematic.pdf" document on the lab GitHub page (https://github.com/namboodirilab/B-CALM/blob/main/circuit%20schematic.pdf). This low cost allows us to control many behavioral experiments in parallel at an affordable price and keeps the system affordable for smaller labs and institutions. The cost of the custom designed head-fixed setup parts is shown in https://github.com/namboodirilab/B-CALM/blob/main/head_fixed_setup/Head-fixed%20stage%20setup%20overview.pdf, and is specific to two-photon microscopy (Namboodiri et al., 2019; Otis et al., 2017).

A major advantage of our system when compared to other open-source behavioral control systems is that we have already established the software and codebase for numerous associative learning and memory tasks (Table 1, see the detailed descriptions and parameter settings of the possible tasks in the “Task running overview.pdf” document provided on the lab GitHub page https://github.com/namboodirilab/B-CALM/blob/main/Task%20running%20overview.pdf; all tested tasks scripts are available on the lab GitHub page task code folder https://github.com/namboodirilab/B-CALM/tree/main/task%20codes). Further, our system includes a user-friendly GUI for setting experiment parameters and plotting real-time data during the experiment. On the other hand, other systems, owing to their general and flexible design, need to be custom written for each application, or require additional work to establish online data visualization for quickly evaluating task performance. While this is eminently doable, it requires considerable programming experience. In our experience, there are a few major advantages of online visualization of task performance. These are: 1) the GUI provides an animal’s up-to-date performance on any day, and thereby lets the user decide when an animal reaches a certain behavioral criterion, 2) the online visualization of timestamps lets the user immediately verify that the task parameters are correct, and 3) the online visualization across multiple behavior boxes in parallel allows quick testing of new behavioral tasks and rapid online modification of task parameters during initial task development.

The number of external connections of our system is only limited by the number of channels available on an Arduino Mega, which has 54 digital input/output pins, 16 analog input/output pins, and 4 UARTs (hardware serial ports). Considering the large ecosystem of shields currently available for the Arduino, our system can in principle be expanded to control more complex tasks than the ones presented here.

Despite these advantages, our system has some limitations. Currently, we do not directly allow the control of arbitrary hardware, other than those taking a TTL input to turn on or off. Thus, our system is not currently built to play arbitrary auditory or visual stimuli. A possible solution to this problem is to interface with other controllers (e.g., other Arduino boards) that deliver the stimuli. For instance, a recent publication used an Arduino system to control an LED panel to deliver arbitrary visual stimuli (Swanson et al., 2021). Similarly, multiple hobby projects demonstrate the playback of notes in songs using an Arduino microcontroller, thereby providing a general solution for a large space of auditory stimuli. Second, as currently built, we have not interfaced our system with live video recording. This can be fixed by using cameras with external TTL triggers or an application programming interface. Lastly, our system is not currently built to integrate multiple separate microcontrollers/processors for the same task with online feedback control. For instance, Autopilot can interface with multiple Raspberry Pis coordinating together to measure many sensors and control multiple actuators, handle video streams, and perform real-time closed-loop experimental control. While our system can do closed-loop control (e.g., deliver reward upon detection of licks), it is not currently built to do this at scale using multiple data streams.

In conclusion, we present a behavior controller for associative learning and memory tasks in the laboratory. All materials necessary for building our system (hardware and software information), instructions for building, list of possible tasks that can be run, and analysis codes are available on the lab GitHub page (https://github.com/namboodirilab/B-CALM/tree/main/data%20analysis%20codes). Our system can be directly implemented by users studying standard associative learning and memory tasks with minimal adaptations. Expert programmers can design arbitrarily complex experiments using our general programming logic since all codes are open source. Hence, we believe that our system is a valuable addition to the open-source toolkit available to a behavioral neuroscientist (White et al., 2019).

Acknowledgments:

We would like to thank Ahmed Morsi and Vincent Lee for help with collecting some behavioral data. We would like to thank Garret Stuber in whose lab V.M.K.N. initially developed the system for controlling Pavlovian conditioning. This work was funded by grants from the National Institute of Health (R00MH118422, R01MH129582, R01AA029661) and the Scott Alan Myers Endowed Professorship to V.M.K.N. The authors declare no conflict of interest.

References:

Ainslie G, Herrnstein RJ (1981) Preference reversal and delayed reinforcement. Animal Learning & Behavior 9:476–482.

Ainslie GW (1974) Impulse control in pigeons. J Exp Anal Behav 21:485–489. [PubMed: 16811760]

Akam T, Lustig A, Rowland J, Kapanaiah SKT, Esteve-Agraz J, Panniello M, Marquez C, Kohl M, Kätzel D, Costa RM, & Walton M (2022). Open source, Python-based, hardware and software for controlling behavioural neuroscience experiments. eLife. 11:e67846 [PubMed: 35043782]

Belsey PP, Nicholas MA, Yttri EA (2020) Open-Source Joystick Manipulandum for Decision-Making, Reaching, and Motor Control Studies in Mice. eNeuro 7.

Brunton BW, Botvinick MM, Brody CD (2013) Rats and humans can optimally accumulate evidence for decision-making. Science 340:95–98. [PubMed: 23559254]

Burgess CP, Lak A, Steinmetz NA, Zatka-Haas P, Bai Reddy C, Jacobs EAK, Linden JF, Paton JJ, Ranson A, Schröder S, Soares S, Wells MJ, Wool LE, Harris KD, Carandini M (2017) High-Yield Methods for Accurate Two-Alternative Visual Psychophysics in Head-Fixed Mice. Cell Rep 20:2513–2524. [PubMed: 28877482]

Burke DA, Jeong H, Wu B, Lee SA, Floeder JR, Namboodiri VMK (2023) Few-shot learning: temporal scaling in behavioral and dopaminergic learning.

Chung S-H, Herrnstein RJ (1967) Choice and delay of reinforcement. J Exp Anal Behav 10:67–74. [PubMed: 16811307]

Crouse RB, Kim K, Batchelor HM, Girardi EM, Kamaletdinova R, Chan J, Rajebhosale P, Pittenger ST, Role LW, Talmage DA, Jing M, Li Y, Gao X-B, Mineur YS, Picciotto MR (2020) Acetylcholine is released in the basolateral amygdala in response to predictors of reward and enhances the learning of cue-reward contingency. eLife 9:e57335. [PubMed: 32945260]

D'Ausilio A (2012) Arduino: a low-cost multipurpose lab equipment. Behav Res Methods 44:305–313. [PubMed: 22037977]

Elliott C, Vijayakumar V, Zink W, Hansen R (2007) National Instruments LabVIEW: A Programming Environment for Laboratory Automation and Measurement. JALA: Journal of the Association for Laboratory Automation 12:17–24.

Fanselow MS, Wassum KM (2016) The Origins and Organization of Vertebrate Pavlovian Conditioning. Cold Spring Harb Perspect Biol 8:a021717.

Ghazizadeh A, Ambroggi F, Odean N, Fields HL (2012) Prefrontal Cortex Mediates Extinction of Responding by Two Distinct Neural Mechanisms in Accumbens Shell. J Neurosci 32:726–737. [PubMed: 22238108]

Goldstein RZ, Volkow ND (2002) Drug addiction and its underlying neurobiological basis: neuroimaging evidence for the involvement of the frontal cortex. Am J Psychiatry 159:1642–1652. [PubMed: 12359667]

Gordon-Fennell A, Barbakh JM, Utley M, Singh S, Bazzino P, Gowrishankar R, Bruchas MR, Roitman MF, Stuber GD (2023) An Open-Source Platform for Head-Fixed Operant and Consummatory Behavior.

Herrnstein RJ (1961) Relative and absolute strength of response as a function of frequency of reinforcement,. J Exp Anal Behav 4:267–272. [PubMed: 13713775]

Jaramillo S, Zador AM (2011) The auditory cortex mediates the perceptual effects of acoustic temporal expectation. Nat Neurosci 14:246–251. [PubMed: 21170056]

Jeong H, Taylor A, Floeder JR, Lohmann M, Mihalas S, Wu B, Zhou M, Burke DA, Namboodiri VMK (2022) Mesolimbic dopamine release conveys causal associations. Science 378:eabq6740. [PubMed: 36480599]

K Namboodiri VM, Hobbs T, Trujillo-Pisanty I, Simon RC, Gray MM, Stuber GD (2021) Relative salience signaling within a thalamo-orbitofrontal circuit governs learning rate. Current Biology 31:5176–5191.e5. [PubMed: 34637750]

K Namboodiri VM, Stuber GD (2021) The learning of prospective and retrospective cognitive maps within neural circuits. Neuron.

Kim K-U, Huh N, Jang Y, Lee D, Jung MW (2015) Effects of fictive reward on rat's choice behavior. Sci Rep 5:8040. [PubMed: 25623929]

Laguesse S, Morisot N, Shin JH, Liu F, Adrover MF, Sakhai SA, Lopez MF, Phamluong K, Griffin WC, Becker HC, Bender KJ, Alvarez VA, Ron D (2017) Prosapip1-Dependent Synaptic Adaptations in the Nucleus Accumbens Drive Alcohol Intake, Seeking, and Reward. Neuron 96:145–159.e8. [PubMed: 28890345]

Livneh Y, Ramesh RN, Burgess CR, Levandowski KM, Madara JC, Fenselau H, Goldey GJ, Diaz VE, Jikomes N, Resch JM, Lowell BB, Andermann ML (2017) Homeostatic circuits selectively gate food cue responses in insular cortex. Nature 546:611–616. [PubMed: 28614299]

MacNiven KH, Jensen ELS, Borg N, Padula CB, Humphreys K, Knutson B (2018) Association of Neural Responses to Drug Cues With Subsequent Relapse to Stimulant Use. JAMA Netw Open 1:e186466. [PubMed: 30646331]

Matell MS, Portugal GS (2007) Impulsive responding on the peak-interval procedure. Behav Processes 74:198–208. [PubMed: 17023122]

Matikainen-Ankney BA et al. (2021) An open-source device for measuring food intake and operant behavior in rodent home-cages. eLife 10:e66173. [PubMed: 33779547]

Mohebi A, Pettibone JR, Hamid AA, Wong J-MT, Vinson LT, Patriarchi T, Tian L, Kennedy RT, Berke JD (2019) Dissociable dopamine dynamics for learning and motivation. Nature 570:65–70. [PubMed: 31118513]

Monroe SC, Radke AK (2021) Aversion-resistant fentanyl self-administration in mice. Psychopharmacology (Berl) 238:699–710. [PubMed: 33226446]

Namboodiri VMK, Huertas MA, Monk KJ, Shouval HZ, Hussain Shuler MG (2015) Visually cued action timing in the primary visual cortex. Neuron 86:319–330. [PubMed: 25819611]

Namboodiri VMK, Otis JM, Heeswijk K van, Voets ES, Alghorazi RA, Rodriguez-Romaguera J, Mihalas S, Stuber GD (2019) Single-cell activity tracking reveals that orbitofrontal neurons acquire and maintain a long-term memory to guide behavioral adaptation. Nat Neurosci 22:1110. [PubMed: 31160741]

Nikbakht N, Diamond ME (2021) Conserved visual capacity of rats under red light. eLife 10:e66429. [PubMed: 34282724]

Otis JM, Namboodiri VMK, Matan AM, Voets ES, Mohorn EP, Kosyk O, McHenry JA, Robinson JE, Resendez SL, Rossi MA, Stuber GD (2017) Prefrontal cortex output circuits guide reward seeking through divergent cue encoding. Nature 543:103–107. [PubMed: 28225752]

Pavlov IP (1927) Conditioned reflexes: an investigation of the physiological activity of the cerebral cortex, Conditioned reflexes: an investigation of the physiological activity of the cerebral cortex. Oxford, England: Oxford Univ. Press.

Perry CJ, Zbukvic I, Kim JH, Lawrence AJ (2014) Role of cues and contexts on drug-seeking behaviour. Br J Pharmacol 171:4636–4672. [PubMed: 24749941]

Poddar R, Kawai R, Ölveczky BP (2013) A Fully Automated High-Throughput Training System for Rodents. PLOS ONE 8:e83171. [PubMed: 24349451]

Richard JM, Ambroggi F, Janak PH, Fields HL (2016) Ventral Pallidum Neurons Encode Incentive Value and Promote Cue-Elicited Instrumental Actions. Neuron 90:1165–1173. [PubMed: 27238868]

Roberts S (1981) Isolation of an internal clock. J Exp Psychol Anim Behav Process 7:242–268. [PubMed: 7252428]

Saunders JL, Wehr M (2019) Autopilot: Automating behavioral experiments with lots of Raspberry Pis.

Schultz BG, van Vugt FT (2016) Tap Arduino: An Arduino microcontroller for low-latency auditory feedback in sensorimotor synchronization experiments. Behav Res Methods 48:1591–1607. [PubMed: 26542971]

Slotnick B (2009) A SIMPLE 2-TRANSISTOR TOUCH OR LICK DETECTOR CIRCUIT. J Exp Anal Behav 91:253–255. [PubMed: 19794837]

Solari N, Sviatkó K, Laszlovszky T, Hegedüs P, Hangya B (2018) Open Source Tools for Temporally Controlled Rodent Behavior Suitable for Electrophysiology and Optogenetic Manipulations. Front Syst Neurosci 12:18. [PubMed: 29867383]

Staddon JER, Cerutti DT (2003) Operant Conditioning. Annu Rev Psychol 54:115–144. [PubMed: 12415075]

Swanson K, White SR, Preston MW, Wilson J, Mitchell M, Laubach M (2021) An Open Source Platform for Presenting Dynamic Visual Stimuli. eNeuro 8:ENEURO.0563-20.2021.

The International Brain Laboratory et al. (2021) Standardized and reproducible measurement of decision-making in mice. eLife 10:e63711. [PubMed: 34011433]

Thorndike EL (1898) Animal intelligence: An experimental study of the associative processes in animals. The Psychological Review: Monograph Supplements 2:i–109.

Toda K, Lusk NA, Watson GDR, Kim N, Lu D, Li HE, Meck WH, Yin HH (2017) Nigrotectal Stimulation Stops Interval Timing in Mice. Curr Biol 27:3763–3770.e3. [PubMed: 29199075]

Vollmer KM, Doncheck EM, Grant RI, Winston KT, Romanova EV, Bowen CW, Siegler PN, Bobadilla A-C, Trujillo-Pisanty I, Kalivas PW, Otis JM (2020) Drug self-administration in head-restrained mice for simultaneous multiphoton imaging. bioRxiv 2020.10.25.354209.

White SR, Amarante LM, Kravitz AV, Laubach M (2019) The Future Is Open: Open-Source Tools for Behavioral Neuroscience Research. eNeuro 6:ENEURO.0223-19.2019.

Woods NI, Stefanini F, Apodaca-Montano DL, Tan IMC, Biane JS, Kheirbek MA (2020) The Dentate Gyrus Classifies Cortical Representations of Learned Stimuli. Neuron 107:173–184.e6. [PubMed: 32359400]

Yttri EA, Dudman JT (2016) Opponent and bidirectional control of movement velocity in the basal ganglia. Nature 533:402–406. [PubMed: 27135927]

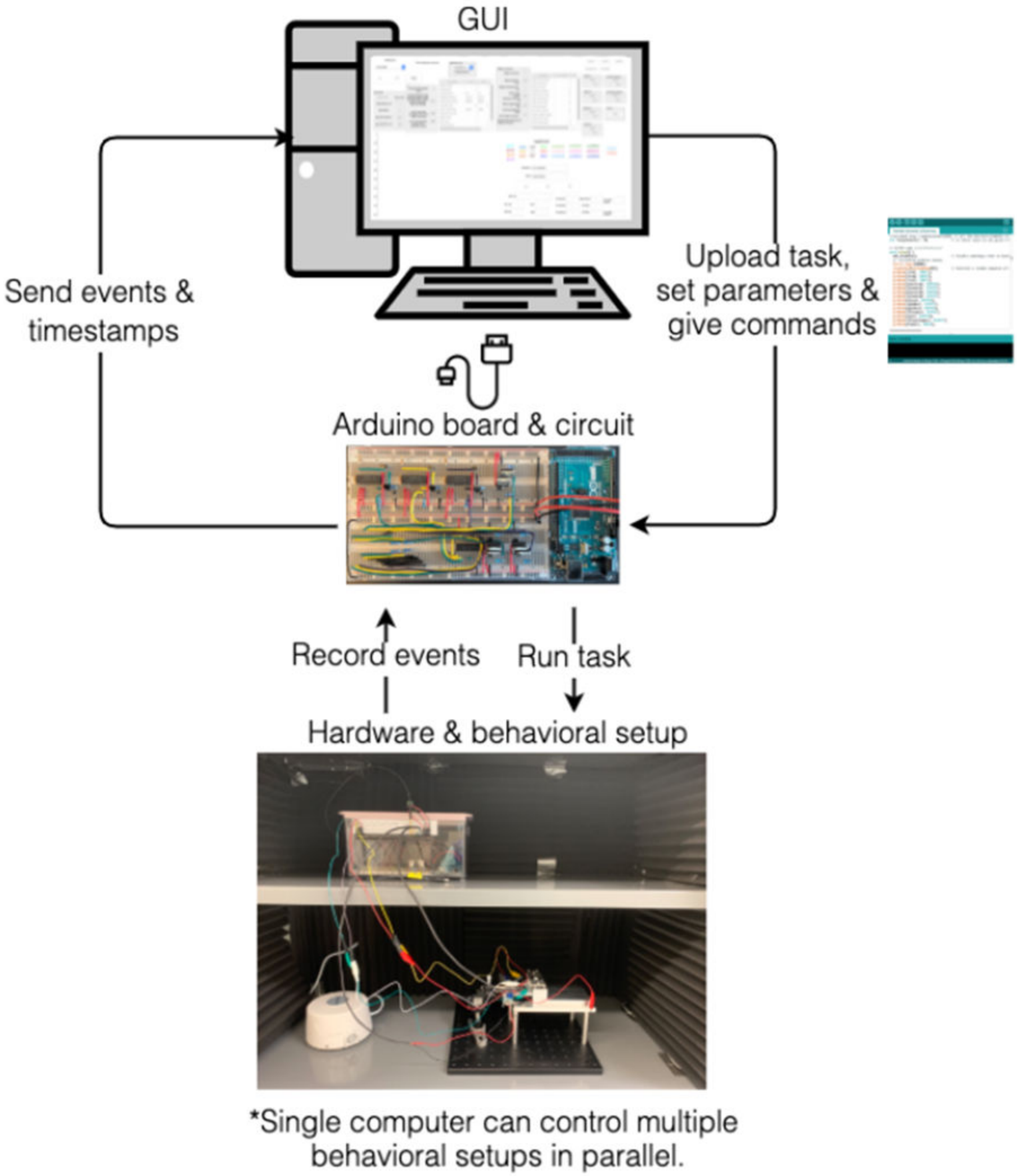

Figure 1. Control setup for B-CALM.

A computer running a MATLAB Graphical User Interfa (GUI) interacts with multiple independently running Arduino Mega microcontrollers (one per behavior box) that each get hooked up to various sensors or actuators to perform standard associative learning and memory tasks.

Figure 2. Screenshot of the GUI.

A. This section selects the Arduino to be controlled, which is displayed as the serial port available to connect (“COM3” in this case). “Link” opens the serial connection between MATLAB and Arduino and “Unlink” disconnects this serial connection. “Experiment Mode” sets the type of experiment that should be currently run and allows automatically uploading the corresponding Arduino compiled hex file to the corresponding Arduino board. **B.** Setting of all the task parameters. Detailed instructions for setting up different parameters is provided in the “Task running overview.pdf” document in the lab GitHub page (https://github.com/namboodirilab/B-CALM/blob/main/Task%20running%20overview.pdf). **C.** Test buttons to test all hardware components. **D.** Example plot of licking patterns in a Pavlovian task. Each lick is shown by a tick mark, each CS1 (CS+) is shown by the green tick mark, and each solenoid (reward) is shown by the orange tick mark. The animal shows anticipatory licking after CS1 trials, but not CS2 (CS−) trials. The lines at the top of the raster indicate the peri-stimulus time histogram of licking for CS1 and CS2 trials, with each one normalized to its maximum value. **E.** Legend for plot shown in **D. F.** File name for saving. The date and time are automatically appended following the text entered here. Additional buttons allow interfacing with the Arduino to send parameters, start the experiment, or stop the experiment (not shown here since it is hidden once task is completed). **G.** Live counters for the various events in the session.

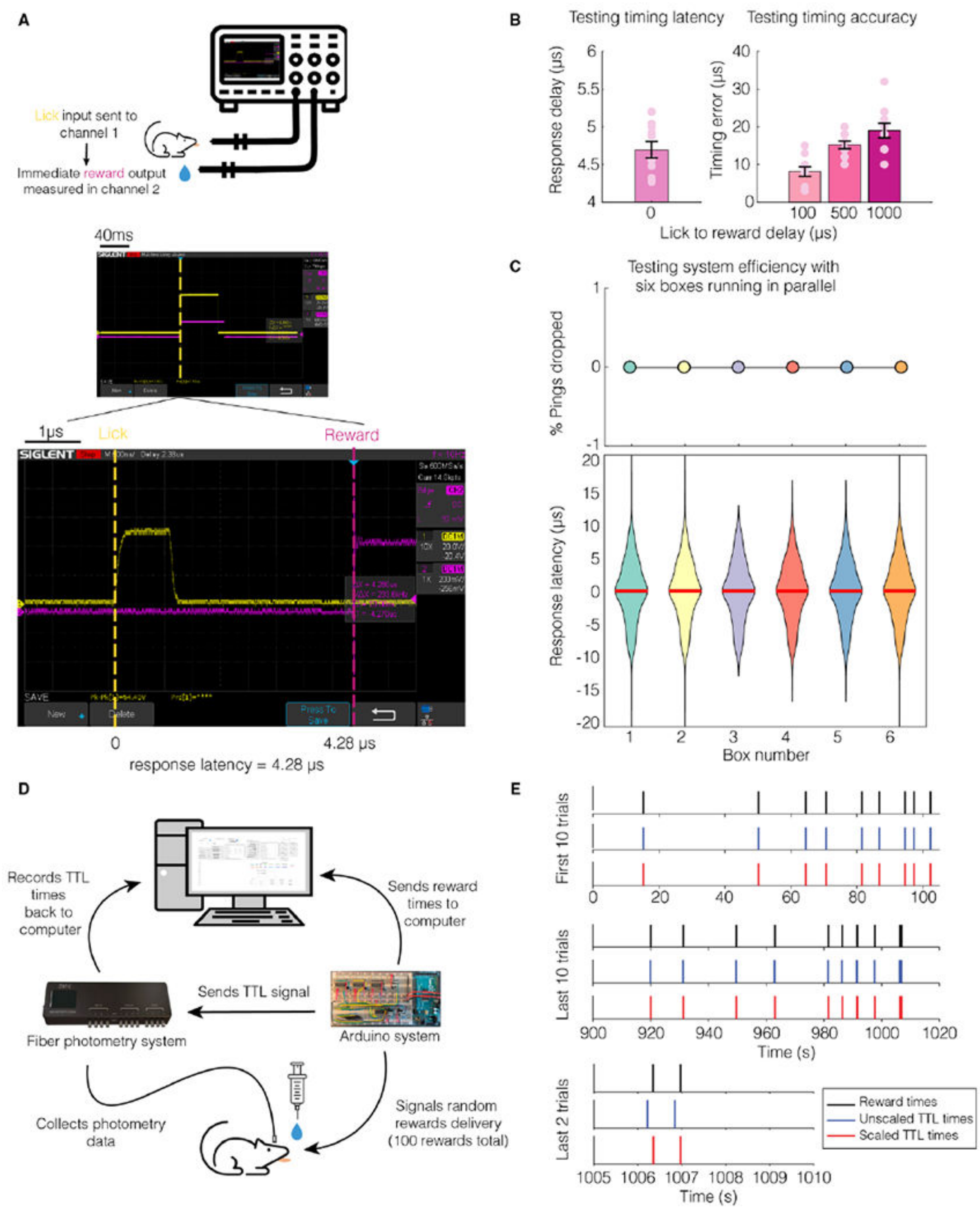

Figure 3. Characterization of the response latency and timing accuracy.

A. Setup for measuring response latency between a lick and the immediately triggered reward. We used an oscilloscope to measure the latencies. On low zoom (middle), both lick and reward produce stable square pulses. However, on high zoom (bottom graph), the lick event produces short pulses of electrical activity immediately after lick detection due to noise. The latency to reward was measured from the first lick induced pulse as shown here. **B.** Response latency and timing accuracy when reward was triggered 0, 100, 500 or 1000 μs after lick detection. Error bars are standard error of the mean. **C.** System does not drop any pings when running six boxes in parallel and has low response latencies across all boxes. **D.** Schematic for synchronization of behavioral control with an example external acquisition system (here, Doric fiber photometry console). We send a TTL output from B-CALM for any set of aperiodic behavioral events (here, reward deliveries) as a TTL input to the external acquisition system. Thus, the timestamps for the same events are recorded both by the Arduino Mega clock in B-CALM and the external acquisition system clock. These events can then be aligned post-hoc using a linear transformation. **E.** Example timestamps from the same TTL events recorded on the B-CALM clock (black lines), external acquisition system clock (blue lines), and after transforming the external acquisition clock timestamps to match the B-CALM clock (red lines). Over the period of ~ 1000s, the two clocks drift from each other, as can be seen the timestamps from the last two trials (bottom row). The synchronization is thus necessary to align events across both clocks.

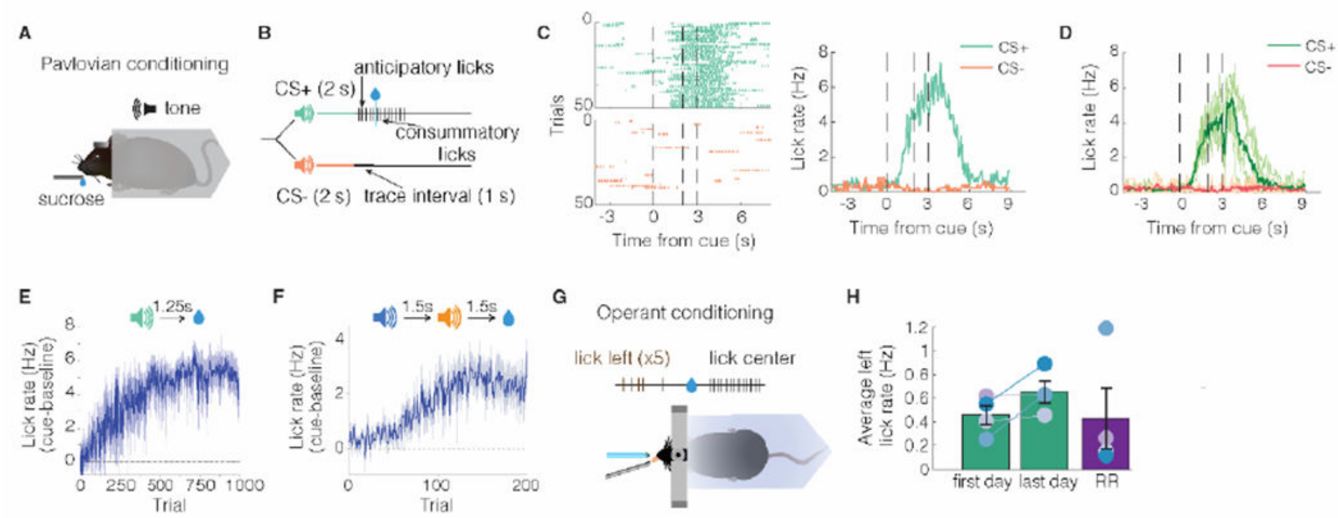

Figure 4. Associative learning and memory tasks for Pavlovian (A-F) or operant (G-H) learning in head-fixed mice.

A. Schematic of head-fixed mouse performing a Pavlovian conditioning task. All error bars or shaded errors are the standard error of the mean. **B.** Timeline of differential trace conditioning. **C.** Raster plot of individual licks during CS+ and CS− trials, showing the trial-averaged lick rate from one example mouse. **D.** Average PSTH across all animals (n=5), with lighter colored lines showing the data from individual animals, showing anticipatory licking only after CS+ cue before reward delivery (paired t test t = 11.6235, two-tailed $p = 3.13 \times 10^{-04}$). **E.** Learning curve for a separate cohort of animals trained on a trace conditioning task with the cue lasting 0.25 s, and a trace interval of 1 s. **F.** Learning curve for another cohort of animals trained on a sequential conditioning task (reproduced from control animals shown in Figure 6 of Jeong et al., 2022). **G.** Task schematic for operant conditioning. Mice had to lick on a left lick tube to obtain rewards from the center lick tube. **H.** The average operant lick rate (Hz) during fixed ratio 5 schedule on first day and last day (paired t test t=−2.0588, one-tailed p = 0.0658), or a single random ratio session.

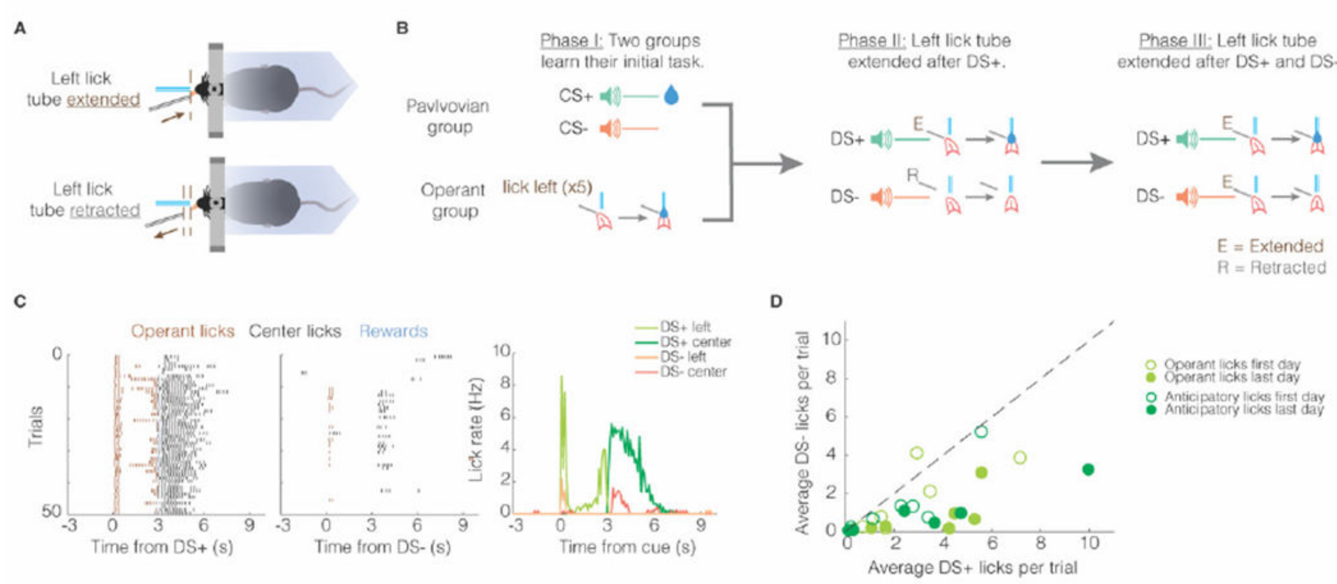

Figure 5. Discrimination learning task.

A. For the discrimination learning task, the left lick tube was retractable and could be in either a retracted or an extended position. **B.** Three phases of training during the discrimination learning task. In the first phase, animals started in either Pavlovian conditioning (n=2 mice) or operant conditioning (n=4 mice). In the second phase, the left lick tube was extended and available for licking only during the cue-outcome delay (3 s) after the discriminative stimulus predictive of reward (DS+). Licking on the left lick tube during this 3s period delivered reward 3s after DS+ onset. In the third phase, the left lick tube was available after both DS+ and DS−. Licking after DS− resulted in no rewards. **C.** Raster plot and peri-stimulus time histogram (PSTH) for operant left licking and Pavlovian/ consummatory center licking from one example animal (the animal was required to make 3 operant licks to obtain reward after DS+). **D.** Average operant licking after DS+ and DS− trials. All animals show more operant licking to DS+ than DS− (paired t test t(5) = 4.3411, one-tailed p = 0.0037).

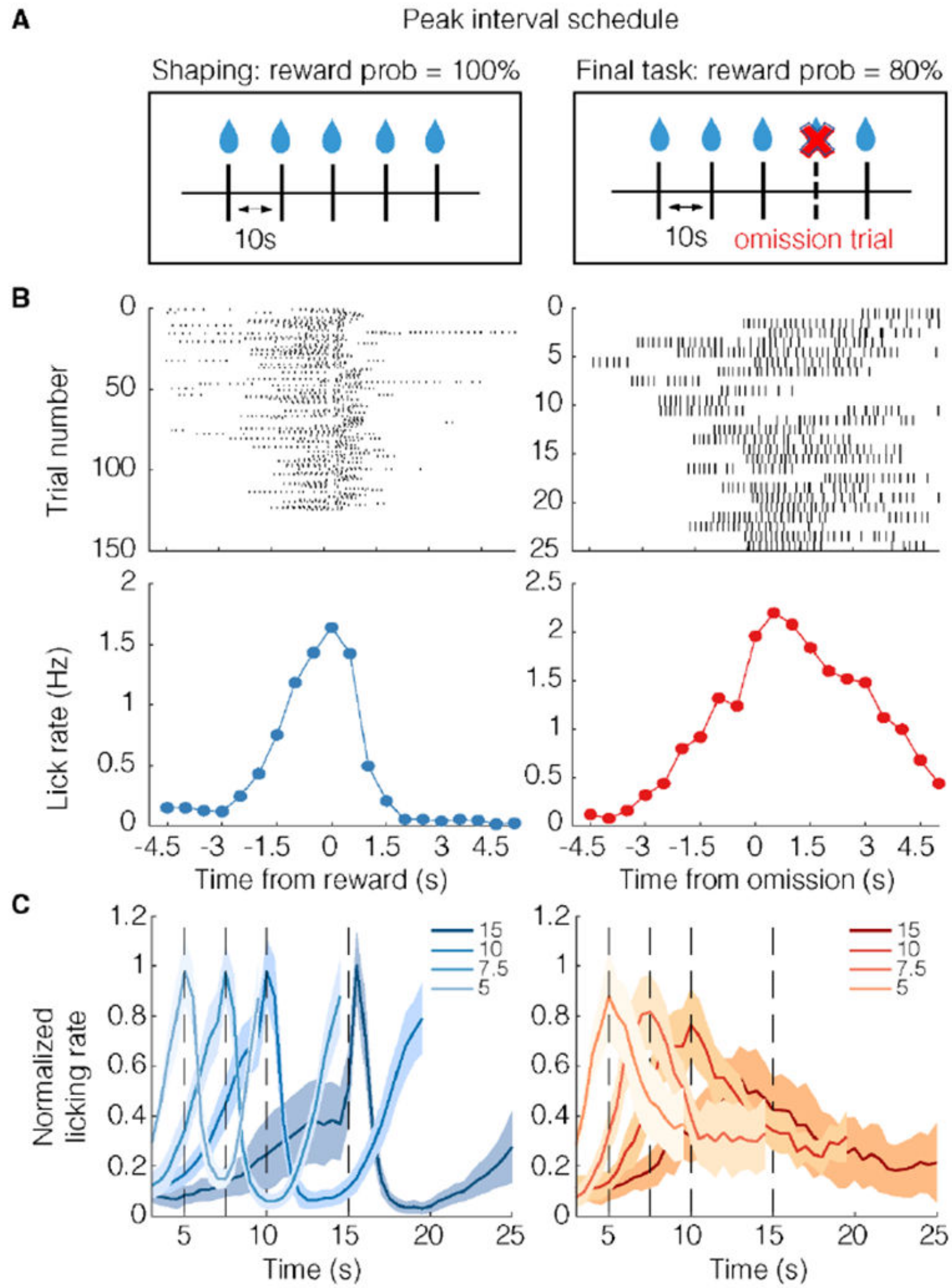

Figure 6. Peak interval timing task.

A. Schematic for the peak interval task. During initial shaping, rewards were given 10 s apart from each other at 100% probability. In the final task, 20% of the rewards were omitted, allowing a chance to measure whether licking “peaked” at the expected reward time (which would signal that animals learn the expected reward time). **B.** Data from an example animal. (Left) Licking time locked to reward receipt, showing anticipatory licking prior to reward receipt. (Right) Licking time locked to reward omission, showing licking around expected reward time. Peak lick rate occurs close to reward omission time. **C.**

Lick rate normalized to peak lick rate averaged across all animals (n=5) for reward receipt (left) and omission (right) for 4 different time intervals. Animals showed clearly distinct licking patterns for each of the four intervals. Indeed, they exhibited unbiased timing during omission trials for 5 s ($t = -0.5349$, two-tailed $p = 0.5989$), 7.5 s ($t = -0.3688$, two-tailed $p = 0.715$), and 10 s ($t = -0.6893$, two-tailed $p = 0.4947$); though, licking peaked with an earlier bias for the 15 s interval ($t = -1.4968$, two-tailed $p = 0.1398$).

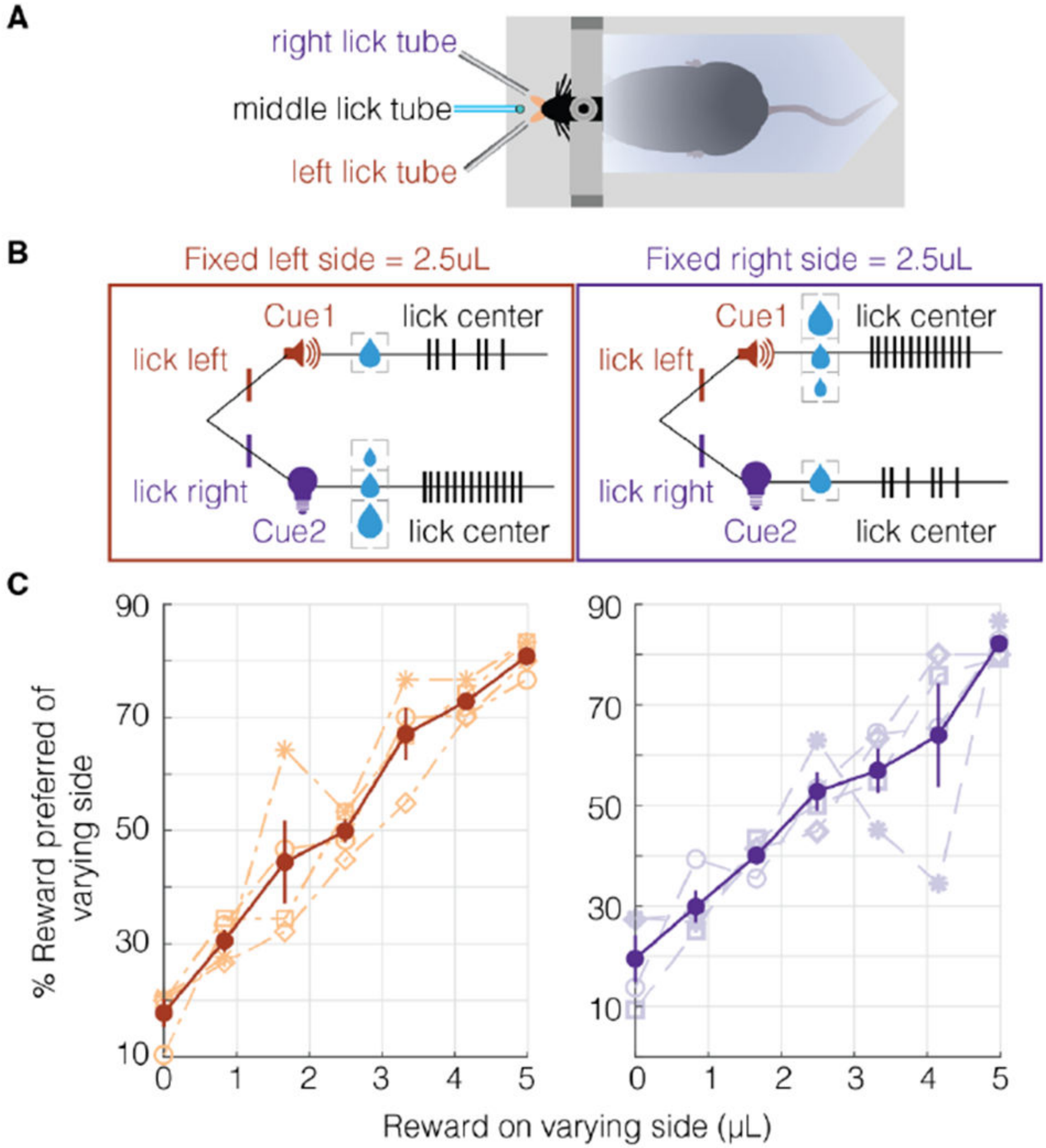

Figure 7. Two-option choice task.

A. Schematic of hardware setup for the two-option choice task. Animals could lick to either the left or right lick tube to obtain corresponding rewards in the middle lick tube. **B.** As a proof of principle and validation, we varied reward magnitude associated with licking either the left or the right side to test whether animals chose both at an equal proportion when the corresponding reward magnitudes were equal. To control for side preference, we did two separate experiments per mouse: one in which reward associated with left lick tube was fixed and the other in which reward associated with the right was fixed. **C.** Proportion of trials in a block (last 10 trials of each condition) that animals chose for the side with varying reward (when either the left or right side was the fixed reward side). Dark line shows average across animals and light lines show individual animals (n=4). For left-side fixed trials, when the right-side reward magnitude equals zero, the percentages of choosing the right side for individual animals were 15.9220%, 25.5296%, 19.0540%, and 20.0412%. The slope for the percentage of choosing right side as right-side reward magnitude increases for individual animals were 12.4511, 12.8431, 12.7527, and 12.9779 (paired t-test t (3) = 114.1708, one-tailed p = 1.4814x10^{-6}). For right-side fixed trials, when the left side reward magnitude equals zero, the percentages of choosing the left side for individual animals were 21.8771%, 25.0001%, 19.7624%, and 13.6347%. The slope for the percentage of choosing the left side as its reward magnitude increases for individual animals were 12.1770, 8.5213, 12.3994, and 13.9000 (paired t-test t (3) =10.2881, one-tailed p = 0.0020). Thus, the choices of the animals were sensitive to the reward magnitudes and there was little side bias in choices.

Author Manuscript

Author Manuscript

Author Manuscript

Table 1.

Short summary of possible tasks capable of being run by B-CALM. See codes and task descriptions in the Github page linked above. Any of the operant action listed can be provided as an input to BCALM using a TTL pulse that turns on at the moments of action performance.

Task	Max # of different stimuli	Stimulus Type	Operant Action	Inter-trial interval
Pavlovian delay/trace conditioning	4	sound/light/both	N/A	exponentially distributed; uniformly distributed; “trial-less” (Jeong et al., 2022)
Operant conditioning	N/A	N/A	lick/nosepoke/lever press	fixed/variable ratio; fixed/variable interval; progressive ratio
Discrimination learning	4	sound/light/both	lick/nosepoke/lever press	exponentially distributed; uniformly distributed
Random rewards	N/A	N/A	N/A	Exponentially or uniformly distributed inter-reward intervals
Interval timing task	N/A	N/A	N/A	fixed inter-reward interval with omissions
Ramp timing task	4	sound/light/both	lick/nosepoke/lever press	exponentially distributed; uniformly distributed (Namboodiri et al., 2015)
Choice tasks (delay/probability/effort discounting; two-armed bandit)	2	sound/light/both	lick/nosepoke/lever press	exponentially distributed; uniformly distributed
Matched-to-sample task	4+4 (second cues)	sound/light/both	lick/nosepoke/lever press	exponentially distributed; uniformly distributed

Table 2.

Summary table for the cost of one BCALM controller system that can reproduce the behavioral results presented here. See https://github.com/namboodirilab/B-CALM/blob/main/head_fixed_setup/Head-fixed%20stage%20setup%20overview.pdf for additional parts unrelated to the controller.

System	Item Name	Vendor/manufacturer part #	Quantity per system	Unit Cost ($)
Single processor	Arduino Mega 2560 board	Arduino A000067	1	40.3
Circuit board	Circuit breadboard	BB1660T	1	15
Reward delivery (1)	Solenoid	Parker 003-0257-900	1	108
	5.6 kΩ resistor	MBB02070C5601FCT00	1	0.27
	TIP120	STMicroelectronics TIP120	1	0.75
	Single diode	1N4001DITR-ND	1	0.22
Lick detector (3)	10 MΩ resistor	YAGEO CFR-25JB-52-10M	6	0.1
	47 kΩ resistor	Vishay Beyschlag/Draloric/BC Components MBB02070C4702FCT00	3	0.27
	10 kΩ resistor	Vishay Beyschlag/Draloric/BC Components MBB02070C1002FRP00	3	0.03
	Reed relay	Excel Cell Electronics EDR2D1A0500Z	3	3.19
	2N222 transistor	MCIGICM 3145241	3	0.035
	Schmitt trigger	TEXAS INSTRUMENTS CD40106BE	1	0.933
	2.2 kΩ resistor	Vishay Beyschlag/Draloric/BC Components MBB02070C2201FCT00	2	0.26
	Power MOFSET	Adafruit 355	2	1.75
Speaker (2)	Resistor kit	E-Projects A-0005-A02	1	16.66
	Large enclosed piezo w/ Wires	Adadruit 1739	1	0.95
	Small enclosed piezo w/ Wires	Adadruit 1740	1	0.95
Light (2)	200 Ω resistor	From the above resistor kit	2	NA
	Pre Wired-LED Diodes Light 5mm-Blue	ToToT PWDIODE-30-Blue	2	0.253
Power Supply	24V/2A Power adapter	Tri-Mag, LLC L6R48-240	1	19.11
	12V/2A power adapter	Tri-Mag, LLC L6R24-120	1	9.74
	9V/1A power adapter	Tri-Mag, LLC L6R12H-090	1	7.62
	Female DC power adapter	Adafruit 368	2	2.00
	USB A-B 2.0 Cable	Qualtek 3021001-03	1	2.63
Necessary circuit elements	Solid-core wire spool	Adafruit 290	5 colors x 25 ft	2.95
	Wire-Wrapping 30AWG wires	CECOMINOD035080	1	10.99
	Jumper wire kit (M to M)	EDGELEC ED-DP_L15_M-M_120pcs	1 set per length	7.49
	Screw terminal block	Molex 0393570006	7	1.86
	Header pin strip	Harwin M20-9990646	8	0.165
	Alligator clips	MCIGICM test lead-10	4	0.499

System	Item Name	Vendor/manufacturer part #	Quantity per system	Unit Cost ($)
Enclosure for circuit setup	Plastic box	14090MDT	1	12.48
Software	MATLAB license	Mathworks	1	variable
Approximate total cost for a complete controller system used for obtaining the results presented here				$303 + MATLAB license cost