BehaviorDEPOT: a tool for automated behavior classification and analysis in rodents

Christopher J Gabriel[1,2], Zachary Zeidler*[2], Benita Jin*[1,2], Changliang Guo[3], Anna Wu[2], Daniel Aharoni[3], Scott A. Wilke[4], and Laura A. DeNardo[2,#]
*equal contribution

Author Affiliations: [1]UCLA Neuroscience Interdepartmental Program, [2]Department of Physiology, [3]Department of Neurology, [4]Department of Psychiatry, David Geffen School of Medicine, University of California Los Angeles

[#]Correspondence: Dr. Laura DeNardo, 10833 Le Conte Ave, CHS 77-200A, David Geffen School of Medicine at UCLA, Los Angeles, CA 90095 ldenardo@ucla.edu

Abstract

Quantitative descriptions of animal behavior are essential to understand the underlying neural substrates. Fear conditioning in rodents is a widely used assay that allows neuroscientists to probe the neural mechanisms of memory. To date, quantification of freezing behavior, a proxy for fear memory strength, is usually performed by hand or with expensive and inflexible commercial software. To overcome these barriers, we developed BehaviorDEPOT (DEcoding behavior based on POsitional Tracking), a MATLAB-based application containing six independent modules. The Experiment Module runs fear conditioning experiments using an Arduino-based design that interfaces with commercial hardware. The Analysis Module classifies freezing and analyzes spatiotemporal behavioral statistics in user-defined ways. The remaining modules can develop custom classifiers. Of note, the Inter-Rater Module establishes reliable 'ground-truth' human labels, making it broadly useful for scientists developing classifiers with any application. BehaviorDEPOT provides a simple, flexible, automated pipeline to move from pose tracking to reliably quantifying task-relevant behaviors.

Introduction

A central goal of neuroscience is to discover relationships between neural activity and behavior. Discrete rodent behaviors represent outward manifestations of cognitive and emotional processes. It is critical to classify such behaviors rapidly and reliably with high spatiotemporal precision. The recent explosion of techniques for manipulating or recording activity in neural circuits is advancing behavioral neuroscience at an unprecedented pace[1]. However, a major bottleneck has been aligning these data with coincident behavior, especially for laboratories without established expertise in this area. Much of this work has been done manually, but automated detection of freely moving animal behaviors is faster, expands the parameter space that can be explored and removes human biases. The standardization promised by such methods also enhances the rigor and reproducibility of results across research groups. However, these analyses can be technically challenging for researchers to develop in house.

Recently, behavioral neuroscientists have embraced machine learning algorithms for tracking animal positions across time and space[2]. Free and easy-to-use software packages such as DeepLabCut (**DLC**) allow even inexperienced users to train deep neural networks that automatically track user-defined points on an animal's body[3]. The static arrangement of those points at any moment represents a pose. The challenge then is to classify temporal pose sequences as discrete behaviors that represent useful information about the cognitive or emotional state of the animal. This represents a major challenge for two main reasons: 1) developing automated classifiers typically requires advanced programming skills and 2) developing robust classifiers requires accurate ground-truth definitions of behaviors, but human annotations are error-prone and unreliable.

Freezing in response to fearful stimuli is a naturalistic behavior that can help rodents avoid predation. This widely studied behavior is commonly measured as a proxy for the strength of the animal's fear memory[4]. Many labs currently score freezing manually, a time-consuming, laborious, and error-prone process. Others use commercially available software (e.g., FreezeFrame and VideoFreeze) that can be prohibitively expensive and that can fail to detect freezing when an animal is wearing commonly used headgear for *in vivo* neurophysiology. Moreover, manual scoring and commercial software typically reports when the animal froze, but not where it froze. Quantifying spatial statistics including freezing location (e.g., corner vs. center of a well-lit arena) or distance traveled can provide important layers of information about the animal's emotional and physical status.

To overcome these challenges, we developed an open source software package for behavior **DE**coding based on **PO**sitional **T**racking (BehaviorDEPOT) that is robust, flexible and easy to use. BehaviorDEPOT is a MATLAB-based application comprising six independent modules (Figure 1). With its graphical user interface (GUI), no coding experience is required to use BehaviorDEPOT, but experienced users can tailor the software to their liking. As an initial use case, we have built modules around measuring freezing behavior. The Experiment Module can run fear conditioning experiments using an Arduino-based protocol that is compatible with commercially available shockers, arenas, and lasers for optogenetics. The Analysis Module imports behavior videos and accompanying pose tracking data and classifies freezing behavior on a frame-by-frame basis. It then performs batched, automated freezing analysis for any experimental design (e.g., cued fear conditioning or optogenetic stimulation). Vectorized data is saved in MATLAB data arrays that facilitate additional user-defined automated analyses. The Analysis Module also features customizable visualization tools that integrate behavioral classification with spatiotemporal trajectory mapping.

Four additional independent modules guide users through designing custom classifiers in a manner that requires no programming experience. A major hurdle in developing classifiers is settling on a ground truth definition of the behavior of interest. It is difficult, particularly for unskilled raters, to annotate hours of behavior videos with introducing errors or having annotations diverge amongst individuals and across labs. However, several skilled raters, focusing on establishing ground-truth definitions can establish behaviors which can then be more reliably annotated by automated classifiers The Inter-Rater Module simplifies this process by performing automated comparisons of manual annotations from multiple human raters. Taking in frame-by-frame human annotations, it calculates the level of inter-rater agreement and highlights points of disagreement so the raters can refine their annotations until they converge maximally. The Inter-Rater module can function as an independent unit and can thus support classifier development with any application. The additional modules help users perform parameter exploration, optimization, and validation of new classifiers.

BehaviorDEPOT joins a growing collection of open source software packages, each with unique strengths, that classify behavior based on animal tracking. Our goal with BehaviorDEPOT was to create a free and user-friendly tool for automated classification and spatiotemporal analysis of commonly studied behaviors. Although it is customizable, BehaviorDEPOT is geared towards analysis of fear conditioned behaviors. Other open source software packages that transform animal poses into discrete behaviors focus on social behaviors or behavioral syllables that are not necessarily associated with specific emotional or motivational states (MARS[5], SimBA[6], MoSeq[7], B-SOiD[8]). To our knowledge, BehaviorDEPOT is the only open source pose tracking-based freezing classifier that operates via a user friendly GUI. Compared to existing open source freezing detection software[9], BehaviorDEPOT can both implement and analyze fear conditioning experiments. BehaviorDEPOT can perform batched data analyses and has the unique ability to integrate freezing classification with the spatiotemporal trajectory of the animal. This facilitates analysis of assays in which the experimenter may want to quantify not only how much an animal freezes overall, but where and when it freezes.

Results

BehaviorDEPOT functions as a MATLAB application or as a standalone executable file. Through its GUI, users can interact with six independent modules (Figure 1, Figure S1). The Analysis Module smooths pose tracking data and quantifies freezing in addition to a variety of defensive and ambulatory behaviors including escape, running, and walking. Users can implement custom temporal and spatial filters to analyze behaviors in varied experimental designs. The other four modules allow users to create and optimize custom classifiers with little-to-no programming experience.

Development of the BehaviorDEPOT Freezing Classifier

BehaviorDEPOT combines a convolution-based smoothing operation with a low-pass filter to identify periods of freezing (Figure 2A–D), defined as the absence of movement except for respiration. We began by training two different deep neural networks in DeepLabCut. One is based on videos recorded with a machine-learning quality camera at 50 frames per second (fps). This networks tracked 8 points on the body (Figure 2A). The second network is based on videos recorded with a standard webcam at 30fps. On the webcam, lower sensor quality and lower frame rate produces more blur in the recorded images, so we only tracked 4 body parts (nose, ears, tail-base; Figure 2). While it is possible to generate a single DLC network that can estimate the positions of body parts with reasonable precision in both video types, we found that more tailored networks produced lower jitter (i.e., small changes in point placement from frame to frame) around the estimated points. Low jitter proved to be critical for creating the most robust freezing classifier.

DLC produces a list of comma-separated values that contains framewise estimates of the x-y coordinates for designated body parts as well as a likelihood statistic for each estimated point. We first applied a threshold based on DLC likelihood ($p < 0.1$) and performed a Hampel transformation[10] to remove outliers. Then, a LOWESS, local regression smoothing method was applied to the data[11], and sub-threshold tracking values were estimated using surrounding data and spline interpolation (Figure 2B). We calculated statistics based on tracked points including linear velocity, acceleration, and angular velocity. After exploring the parameter space, we empirically determined that thresholding the velocity of a weighted average of 3–6 body parts (depending on the frame rate of the video recording) and the angle of head movements produced the best-performing freezing classifier (Figure 2C). We applied a sliding window to produce a convolved freezing vector in which each value represents the number of freezing frames visible in the window at a given frame. We then applied an adjustable count threshold to convert the convolved freezing vector into the final binary freezing vector (Figure 2D).

We validated the performance of the freezing classifier by comparing it to manual scoring by six trained raters. To ensure our classifiers were robust, our validation data sets included 30,000 frames from 5 videos (6000 frames per video) that were recorded in distinct behavioral chambers under varied lighting conditions. Human ratings were indistinguishable from classifier performance (Figure 2E,F). Accuracy of the freezing classifier was estimated based on classifier precision (number of true positive frames / sum of true positive and false positive frames), recall (number of true positive frames / sum of true positive and false negative frames), F1 score (two times the product of precision and recall / sum of precision and recall) and specificity (number of true negative frames / sum of true negative and false positive frames). Precision and recall quantify the positive predictive value of the classifier against the tendency to produce false positive or false negative errors, respectively. The F1 score is the harmonic mean of the precision and recall is useful as a summary statistic of overall classifier performance. Specificity quantifies the classifier's ability to accurately label true negative values and helps ensure that the classifier is capturing data from only a single annotated behavior. These values indicated that freezing classifiers based on webcams and machine learning quality cameras both had excellent performance.

The Analysis Module

The Analysis Module imports videos and accompanying pose tracking data, smooths the data as described above, and calculates movement statistics based on tracked points (e.g., linear velocity, angular velocity, acceleration). It then analyzes behavior in a framewise manner by implementing custom classifiers that threshold movement and position statistics in unique combinations. In the GUI, users indicate if they want to analyze behaviors during particular time windows or within regions of interest (ROI), and whether they plan to do batched analyses. Depending on their choices, users are led through a series of windows in which they indicate the body parts that they tracked, ROI location, and when time windows of interest occurred. The Analysis Module analyzes behavior according to the user-defined specifications and saves the data in a set of MATLAB structures. The 'Params' structure stores the parameters of video recordings, smoothing, and arena metrics (e.g., ROI size and location). 'Tracking' organizes and stores pose data from DLC. 'Metrics' stores calculated movement and position statistics based on tracked points including linear velocity, angular velocity, and acceleration (Figure 2G). Organizing the data this way facilitates additional automated analyses of the user's choosing. Users can track any points they like, though they may have to adjust and validate the freezing thresholds using our Optimization and Validation modules, described in upcoming sections.

The Analysis Module also generates a series of graphical data representations. Trajectory maps show when an animal was in a particular location and where behaviors occurred. Bout maps indicate when behaviors occurred and for how long. Additional graphs tell the user the amount a particular behavior occurred during presentations of a cue or optogenetic stimulation period. These visual representations help users understand behavioral phenotypes in great spatiotemporal detail and can inform development of further custom analyses using the data arrays that the Analysis Module generates.

Use Case 1: Optogenetics

In commercial freezing detection software, algorithms often fail when a rodent is wearing a patch cord for routine optogenetics experiments. We set out to validate BehaviorDEPOT's utility in an optogenetics experiment. The medial prefrontal cortex (mPFC) plays a well-established role in fear memory retrieval, extinction, and generalization[12]. To validate the utility of BehaviorDEPOT for fear conditioning experiments that use optogenetics, we performed an experiment to examine the role of mPFC in contextual fear memory generalization. While silencing mPFC subregions can promote fear memory generalization in remote memory[13,14], less is known about its role in recent memory generalization. We used an adeno-associated virus (AAV) to express the soma-targeted neuronal silencer stGtACR2[15] bilaterally in the mPFC and implanted optogenetic cannula directly above the AAV injection sites (Figure 3A, Figure S2). We performed contextual fear conditioning (CFC) and then tested animals' fear memory 24 hours later by recording their behavior in the fearful context and then in a novel environment that was not previously paired with shocks (Figure 3B). We estimated the animals' position on each frame using an optogenetics-specific DLC network that tracked 9 points on the animal, including the fiber optic cannula. We then analyzed freezing levels with BehaviorDEPOT. Fear conditioned mice readily froze following shocks during CFC, while nonshocked controls did not (Figure 3D). Silencing mPFC in previously shocked animals significantly enhanced freezing in the novel context but did not affect freezing in the fearful context (Figure 3E,F). mPFC silencing thus produced a significant decrease in the discrimination index in fear conditioned mice (Figure 3G), indicating that mPFC plays a key role in the specificity of recent fear memories. In all analyses, BehaviorDEPOT performance was indistinguishable from a highly trained human rater (Figure 3E–G) indicating that the combination of the optogenetics-specific DLC model and BehaviorDEPOT achieves robust and accurate freezing detection.

Use Case 2: Ca^{2+} imaging with Miniscopes during learned avoidance

Simultaneous neurophysiological and behavioral recordings are commonplace in modern neuroscience. In fields that use associative learning paradigms to model memory and decision making, Ca^{2+} imaging has become widely adopted to investigate the neural mechanisms. There is increasing evidence that neuronal ensembles act as computational units to associate external cues with behavioral responses[16]. To understand how animals respond to cues that predict rewards or threats, it is ideal to measure the neural repertoires that link environmental cues to behaviors in freely moving animals. However, few labs have done so, in part due to the challenges of reliably quantifying complex behaviors and aligning behavioral annotations with physiological data. To date, must studies have focused on spatial or sensory analyses in freely moving animals (e.g., measuring place cells), or separately studied the encoding of cues that predict rewards or threats. To understand the neural correlates of cue responses in a more naturalistic setting, we now face the challenge of correlating ensemble activity with complex, freely moving behaviors that are modulated by temporal (e.g. conditioned tones) and spatial (e.g. safety zone) factors. BehaviorDEPOT directly addresses these deficits, producing vectorized data structures containing spatial information (e.g animal location), temporal information (e.g. onset of conditioned cues), and the time and location at which discrete behaviors occurred that can be instantly aligned with physiolgical recordings. We recorded ensembles of neurons in freely behaving animals using head-mounted microendoscopes (UCLA Miniscopes). We imaged neural responses during platform mediated avoidance (PMA), in which a rodent must step onto a safety platform to avoid a signaled shock. We recorded Ca^{2+} transients in mPFC using a UCLA Miniscope while simultaneously recording behavior using a new open source USB camera, the UCLA MiniCAM.

The MiniCAM is an open source behavioral imaging platform that natively integrates and synchronizes with open source UCLA Miniscope hardware and software (Figure 3A). It is composed of an M12 optical lens mount, a custom printed circuit board housing a CMOS image senor and supporting electronics, LED illumination ring, and 3D printed case. The MiniCAM is powered and communicates over a single coaxial cable that can be up to 15 meters long. The coaxial cable connects to a Miniscope data acquisition board (DAQ) which then connects over USB3 to a host computer. A range of commercial M12 lenses can be used to select the view angle of the camera system. The image sensor used is a 5MP CMOS image sensor (MT9P031I12STM-DP, ON Semiconductor) with 2592 x 1944 pixel resolution and a full resolution frame rate of approximately 14FPS. For this application, the MiniCAM's pixels were binned and cropped to achieve 1024X768 pixels at approximately 50FPS. The optional LED illumination ring uses 16 adjustable red LEDs (LTST-C190KRKT, Lite-On Inc., 639nm peak wavelength) for illumination in dark environments (Figure 4A). We trained a separate DLC network for videos of animals wearing Miniscopes recorded with MiniCAMs. Our network tracked 8 points on the body (ears, nose, midback, hips, tailbase, and tail) and the Miniscope itself (Figure 4B). Using the output of this network, BehaviorDEPOT classifed freezing with high fidelity in a manner that was indistinguishable from highly trained users (Figure 4C).

In certain assays, it is advantageous to integrate spatial tracking with behavioral classification. To validate the utility of BehaviorDEPOT for analyzing behavior in user-defined spatiotemporal windows, we analyzed mouse behavior and the underlying neural activity in prefrontal cortex during Platform Mediated Avoidance (PMA). In these experiments, we first injected AAV-syn-GCaMP7f into prelimbic cortex (PL) and then implanted a gradient refractive index (GRIN) lens above the virus injection site. After recovery, animals were trained in PMA. In this task, animals hear 3 baseline tones (4kHz, 30s) followed by 9 tones that co-terminate with a mild foot shock (0.15mA, 2s). An acrylic platform occupies 25% of the electrified grid floor, providing a safe place to avoid the shock. The following day, animals are presented with 6 unreinforced tones and we measure the resulting avoidance and freezing responses. (Figure 4D).

We used BehaviorDEPOT to analyze behavior during PMA so we could align it to the underlying neural activity. The Analysis Module automatically produces maps that make it easy and fast to assess the spatiotemporal characteristics of rodent behavior. In our representative example the color-coded trajectory and freezing locations (denoted as black squares) converge on the platform at the end of the session, indicating the mouse indeed learned to avoid shocks (Figure 4E). The vectorized session data from the Experiment Module (e.g. tone times, shock times) and behavioral data from the Analysis module can be easily integrated for additional analyses. We visualized the temporal dynamics of learning by overlaying fraction time on the platform and freezing on the tone periods (Figure 4F). We also used BehaviorDEPOT to produce summary data, showing that mice readily learned and remembered the cue-avoidance association as expected (Figure 4G).

During a retrieval session, we performed simultaneous recordings of neural activity using a UCLA Miniscope and behavior using a UCLA MiniCAM (Figure 4H). Using MIN1PIPE[17], we extracted and processed neural signals from 131 mPFC neurons. We then determined whether individual neurons encoded specific behaviors that we had quantified using BehaviorDEPOT. We computed a receiver operator characteristic (ROC) curve that measures a neurons stimulus detection strength over a range of thresholds (Figure 4I,J). We identified a number of neurons that were modulated by freezing and avoidance that were organized in a salt and pepper manner in mPFC (Figure 4K). Most neurons exhibited task relevant behavior and were specific for either freezing or avoiding on the platform, or the combination of both (Figure 4L).

The Inter-Rater Module

A major challenge in behavioral neuroscience is establishing reliable ground truth definitions of behavior. Even when using automated behavioral classification systems like BehaviorDEPOT, it is critical to have reliable behavioral annotations from multiple human raters to validate the performance of the classifier. The Inter-Rater Module helps users establish reliable ground truth behavioral definitions in two ways. First, the inter-rater module imports manual behavioral scoring from human raters and automatically compares the scores on a frame-by-frame basis. It calculates the level of agreement and reports points of disagreement both numerically and graphically. In response to the initial output of the Inter-Rater Module, human raters can discuss the points of disagreement and modify their manual ratings accordingly. The automated features of the Inter-Rater module make it fast and easy to perform iterative comparisons of manual annotations, interleaved with human updates, until a satisfactory level of agreement is achieved.

The Inter-Rater module prompts the user to select a file directory containing one or more human annotation files and automatically imports the data into MATLAB. The user is then prompted to select the annotations to include in the analysis, the reference dataset, and the behavior of interest. After all parameters are set, the module compares each set of annotations to the reference, scoring the annotations frame-by-frame as true positive, true negative, false positive, or false negative for each rater. These values are first used to calculate percent overlap and percent error between all raters. Precision, recall, specificity, and F1 score are calculated and reported for each rater relative to the chosen reference. Additionally, visualizations of frame-by-frame percent agreement and user-by-user disagreement are automatically generated to assist identifying areas of conflict between users (Figure 5A). We developed the BehaviorDEPOT freezing classifier based on the averaged ratings of three highly trained users. Here, the Inter-Rater Module demonstrates visualizations of agreement levels for three highly trained raters and a novice freezing rater (Figure 5B). By illustrating frames with high level of disagreement, Inter-Rater can help labs train new human annotators while building ground truth definitions for new classifiers.

The Data Exploration Module

The Data Exploration Module is designed to help refine existing classifiers as well as generate entirely new ones. When developing behavior classifiers, it is advantageous to explore the parameter space of positional tracking data and identify the metrics that track most closely with a behavior of interest. The Data Exploration Module allows users to explore the metrics that BehaviorDEPOT calculates based on tracked points and determine whether chosen metrics separate behavior data accurately, using human labels as a reference. The chosen metrics are used to generate a generalized linear model (GLM) that estimates the likelihood that the behavior is present in a frame, given different values of the selected metrics (Figure 6A). This module allows users to easily identify and compare metrics that can form the basis of new classifiers or enhance the performance of existing ones.

The Data Exploration Module prompts the user to select a directory containing analyzed data and a human rater file, and the data is then automatically imported into MATLAB. Users can select two of the metrics from the analyzed file (e.g., head velocity, tail angular velocity, etc.) and a behavior label from the human annotations file. The module then creates two data distributions: one containing video frames labeled with the chosen behavior and a second consisting of the remaining video frames (Figure 6B). The module will randomly downsample the larger set to ensure that each distribution contains equal numbers of frames. The module then performs a series of analyses to quantify how reliably chosen metrics align with the behavior of interest. Histograms and boxplot representations of metric values for behavior-containing and non-behavior-containing frames help users identify metric values that reliably segregate with frames containing a behavior of interest (Figure 6C). Chosen metrics then form the basis of a GLM that predicts the likelihood that a frame captures the chosen behavior based on different values of each of the chosen metrics. Metrics that are well-suited for behavior classification contrast with metrics on frames that do not contain the behavior and have a low standard deviation within the behavior set (Figure 6D). Distributions of useful metrics also tend to differ substantially from the total set of frames, especially when compared to frames that do not contain the behavior. The GLM predictions are useful for determining which of the selected metrics best predict the behavior and whether they enhance the predictive value when combined.

The Classifier Optimization Module

The Classifier Optimization Module allows users to rapidly assess how changes in classifier parameters (thresholded metrics) affect the performance of the classifier (Figure 7A). It takes in a BehaviorDEPOT-analyzed tracking file and associated human annotations and prompts the user to select up to two parameters from their chosen classifier (Figure 7B). The user can then input a list of parameter values to test on the classifier. The module then runs the classifier using every combination of test values and calculates the performance (precision, recall, F1 score, and specificity) for each parameter combination. These values are stored in a MATLAB data array and automatically visualized using easy-to-read heatmaps. In building the freezing classifier for BehaviorDEPOT, the histogram plotting function of the Classifier Optimization Module revealed a combination of velocity thresholds for two metrics (averaged head/rear back and head angle) that produced a maximal F1 score. We selected that value as the basis of a subsequent iteration that optimized window width and count threshold the convolution algorithm used. These iterative optimizations produced increasingly higher F1 values for classifier performance (Figure 7C). This module is designed to allow users to optimize parameter values for individual classifiers and can be used in conjunction with the data exploration module to build and test new behavior classifiers in a way that requires no prior coding experience.

The Validation Module

The Validation Module is designed to quickly assess a classifier's predictive quality across distinct video datasets. To ensure that behavioral classifications are robust to different video recording conditions, the Validation Module compares classifier output to trained human raters across frames from multiple videos. The user will initially be prompted to indicate which classifier to evaluate and select a directory containing behavior videos and accompanying BehaviorDEPOT-analyzed tracking data. For each video, the module will categorize each frame as true positive, false positive, true negative, or false negative, using the human data as a reference. Precision, recall, specificity, and F1 score are then calculated and visualized for each video. These statistics are also reported for the total video set by concatenating all data and recalculating performance (Figure 7D).

Discussion

In the spirit of expanding the behavioral neuroscientist's open-access scientific toolkit, BehaviorDEPOT extends the utility of existing popular tools for markerless point tracking (e.g. DLC[3] and LEAP[18]) to create an open source, flexible, reliable, and automated behavioral analysis pipeline. BehaviorDEPOT is primarily discussed regarding spatiotemporal and event-triggered freezing analysis but is well-suited for a variety of locomotive and defensive behaviors. The modules create a complete, graphical interface-based pipeline for fear conditioning experimenters, taking users from performing a fear conditioning experiment, validating and optimizing the pre-designed freezing classifier, then classifying and organizing analyses based on spatial (e.g., zones) and/or temporal (e.g., tone times, optogenetic stimulation times) experimental considerations. Additional modules allow users to create, test, and optimize user-defined behavioral classifiers, extending the utility of BehaviorDEPOT beyond the fear conditioning space.

We demonstrated the utility of BehaviorDEPOT's freezing classifier in a variety of contexts with potential visual confounds: unoperated mice, tethered mice with an optogenetic patch cable, as well as mice wearing Miniscopes. In this way, our freezing classifier overcomes a major hurdle associated with commercially available alternatives. While we highlight the freezing classifier as it has an immediate and wide-spread application, we also provide tools for experimenters to create custom behavioral classifiers. Four complementary modules walk users through creating a consensus classification for human-defined behaviors, exploring the point tracking data to identify the metrics and threshold values that best capture the behavior, and then testing, optimizing, and implementing the user-defined classifier in additional analyses. Moreover, the outputs of BehaviorDEPOT are stored in vectorized formats that allow for immediate alignment with neurophysiological recordings. These tools extend the functionality of

BehaviorDEPOT beyond analysis of fear conditioning to a general utility software that is applicable to a wide range of behavioral neuroscience experiments.

BehaviorDEPOT adds to a growing field of open source software for automated behavioral analyses but fills an increasingly apparent gap in current resources. We integrate markerless point tracking with top-down recordings, enabling behavioral classification with two-dimensional spatial precision in an interface that requires no coding experience. Our software is free and open source, allowing no-cost and unlimited use to scientists everywhere. Existing free and open source software has succeeded in addressing other specific needs. For instance, ezTrack[9] enables robust, Python-based freezing classification without the need of point tracking, but is limited to side-view videos, which constrains spatial analyses. MoSeq[7] combines 3-dimensional video recordings with unsupervised machine learning to identify subsecond behavioral motifs, the complexity and coding-requirements of which may be beyond the scope of experimenters interested in more macro-level behaviors. While BehaviorDEPOT focuses on the defensive and exploratory behaviors of single animals, the field continues to develop additional behavioral classifiers to fit specific scientific needs including multi-animal tracking for analyzing social behaviors[5,6] and grooming behaviors[8].

References

1. Luo, L., Callaway, E. M. & Svoboda, K. Genetic Dissection of Neural Circuits: A Decade of Progress. *Neuron* (2018) doi:10.1016/j.neuron.2018.03.040.
2. Mathis, M. W. & Mathis, A. Deep learning tools for the measurement of animal behavior in neuroscience. *Curr. Opin. Neurobiol.* **60**, 1–11 (2020).
3. Mathis, A. *et al.* DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nat. Neurosci.* **21**, (2018).
4. Ledoux, J. E. Emotion circuits in the brain. *Annu. Rev. Neurosci.* 155–184 (2000).
5. Segalin, C. *et al.* The Mouse Action Recognition System (MARS): a software pipeline for automated analysis of social behaviors in mice. *bioRxiv* 2020.07.26.222299 (2020) doi:10.1101/2020.07.26.222299.
6. Nilsson, S. R. O. *et al.* Simple Behavioral Analysis (SimBA) – an open source toolkit for computer classification of complex social behaviors in experimental animals. *bioRxiv* 2020.04.19.049452 (2020) doi:10.1101/2020.04.19.049452.
7. Wiltschko, A. B. *et al.* Revealing the structure of pharmacobehavioral space through motion sequencing. *Nat. Neurosci.* **23**, 1433–1443 (2020).
8. Hsu, A. I. & Yttri, E. A. B-SOiD: An Open Source Unsupervised Algorithm for Discovery of Spontaneous Behaviors. *bioRxiv* 770271 (2019) doi:10.1101/770271.
9. Pennington, Z. T. *et al.* ezTrack: An open-source video analysis pipeline for the investigation of animal behavior. *Sci. Rep.* **9**, 1–11 (2019).
10. Hampel, F. R. The Influence Curve and its Role in Robust Estimation. *J. Am. Stat. Assoc.* **69**, 383–393 (1974).
11. Cleveland, W. S. LOWESS: A Program for Smoothing Scatterplots by Robust Locally Weighted Regression. *Am. Stat.* **35**, 54 (1981).
12. Giustino, T. F. & Maren, S. The Role of the Medial Prefrontal Cortex in the Conditioning and Extinction of Fear. *Front. Behav. Neurosci.* **9**, 1–20 (2015).
13. Frankland, P. W., Bontempi, B., Talkton, L. E., Kaczmarek, L. & Silva, A. J. The Involvement of the Anterior Cingulate Cortex in Remote Contextual Fear Memory. *Science (80-.).* **304**, 881–883 (2004).
14. Xu, W. & Südhof, T. C. A neural circuit for memory specificity and generalization. *Science (80-.).* **339**, 1290–1295 (2013).
15. Mahn, M. *et al.* High-efficiency optogenetic silencing with soma-targeted anion-conducting channelrhodopsins. *Nat. Commun.* **9**, 4125 (2018).
16. Josselyn, S. A. & Tonegawa, S. Memory engrams: Recalling the past and imagining the future. *Science (80-.).* **367**, (2020).

17. Lu, J. *et al.* MIN1PIPE: A Miniscope 1-Photon-Based Calcium Imaging Signal Extraction Pipeline. *Cell Rep.* (2018) doi:10.1016/j.celrep.2018.05.062.
18. Pereira, T. D. *et al.* Fast animal pose estimation using deep neural networks. *Nat. Methods* **16**, 117–125 (2019).
19. Cai, D. J. *et al.* A shared neural ensemble links distinct contextual memories encoded close in time. *Nature* (2016) doi:10.1038/nature17955.
20. Shuman, T. *et al.* Breakdown of spatial coding and interneuron synchronization in epileptic mice. *Nat. Neurosci.* (2020) doi:10.1038/s41593-019-0559-0.

Author Contributions C.J.G. and B.J. developed the freezing classifier. Z.Z. created MATLAB GUIs. C.J.G., B.J., Z.Z., and A.W. performed manual annotations of behavior. C.J.G. performed data acquisition and analysis of optogenetics experiments and Miniscope recordings. Z.Z. contributed to Miniscope data analysis. C.G. and D.A. designed and fabricated the UCLA MiniCAM and wrote accompanying software. L.A.D., S.A.W., and C.J.G. conceived of experiments. C.J.G., Z.Z., B.J., S.A.W., and L.A.D. wrote the paper. All authors read and edited the paper.

Competing Interests The authors declare no competing interests.

Methods

Animals

Female and male C57Bl6/J mice (JAX Stock No. 000664) were group housed (2–5 per cage) and kept on a 12 hr light cycle. Following behavior conditioning, animals were individually housed until the memory retrieval sessions. All animal procedures followed animal care guidelines approved by the University of California, Los Angeles Chancellor's Animal Research Committee.

Contextual fear conditioning

Mice were handled for 5 days preceding the behavioral testing procedure. The conditioning chamber consisted of an 18cm x 18cm x 30cm cage with a grid floor wired to a scrambled shock generator (Lafayette Instruments) surrounded by a custom-built acoustic chamber. The chamber was scented with 50% Windex. Mice were placed in the chamber and then after a 2 minute baseline period, received 5 0.75mA footshocks spaced 1 minute apart. Mice were removed 1 minute after the last shock. Non-shocked control animals freely explored the conditioning chamber but never received any shocks. The following day, mice were returned to the conditioning chamber and a novel context (different metal floor, scented with 1% acetic acid), separated by a 1 hour interval. Context presentation order on day 2 was counterbalanced across mice.

Platform-mediated avoidance

PMA used the fear conditioning chamber described above, except 25% of the floor was covered with a thin acrylic platform (3.5x4x0.5 inches). During training, mice were presented with 3 baseline 30s 4kHz tones (CS), followed by 9 presentations of the CS that co-terminated with a 2s footshock (0.13mA). The following day, mice were presented with 6 CS in the absence of shocks.

Viruses

AAV1-syn-jGCaMP7f.WPRE (ItemID: 104488-AAV1) were purchased from Addgene and diluted to a working titer of 8.5x10^{12} GC/ml. and AAV1-CamKIIa-stGtACR2-FusionRed (ItemID: 105669-AAV1) were purchased from Addgene and diluted to a working titer of 9.5x10^{11} GC/ml.

AAV injection with Optogenetic Cannula Implant

Adult WT mice were anesthetized with isoflurane and secured to a stereotaxic frame. Mice were placed on a heating blanket and artificial tears kept their eyes moist throughout the surgery. After exposing the skull, we drilled a burr hole above mPFC in both hemispheres (AP+1.8, ML+/-0.3 from bregma). A Hamilton syringe containing AAV-CaMKIIa-stGtACR2-WPRE was lowered into the burr hole and 400nL of AAV was pressure injected into each site (DV -2.25mm and -2.50mm from bregma) at 100nL/min using a microinjector (Kopf, 693A). The syringe was left in place for 10 minutes to ensure the AAV did not spill out of the target region. After injecting the AAV, chronic fiber optic cannula (0.37NA, length=2mm, diameter = 200um) were implanted bilaterally above the injection site and secured to the skull with Metabond (Parkell, S371, S396, S398). After recovery, animals were housed in a regular 12hr light/dark cycle with food and water ad libitum. Carprofen (5mg/kg) was administered both during surgery and for 2d after surgery together with amoxicillin (0.25 mg/mL) in the drinking water for 7d after surgery.

Miniscope Surgery and Baseplating

For Miniscope recordings, all mice underwent two stereotaxic surgeries[19,20]. First, adult WT mice were anesthetized with isoflurane and secured to a stereotaxic frame. Mice were placed on a heating blanket and artificial tears kept their eyes moist throughout the surgery. After exposing the skull, a burr hole was drilled above PL in the left hemisphere (+1.8, -0.4, -2.3 mm from bregma). A Hamilton syringe containing AAV1-Syn-jGCaMP7f-WPRE was lowered into the burr hole and 400nL of AAV was pressure injected using a microinjector (Kopf, 693A). The syringe was left in place for 10 minutes to ensure the AAV did not spill out of the target region and then the skin was sutured. After recovery, animals were housed in a regular 12hr light/dark cycle with food and water ad libitum. Carprofen (5mg/kg) was administered both during surgery and for 2d after surgery together with amoxicillin (0.25 mg/mL) for 7d after surgery. One week later, mice underwent a GRIN lens implantation surgery. After anesthetizing the animals with isoflurane (1–3%) and securing them to the stereotaxic frame, the cortical tissue above the targeted implant site was carefully aspirated using 27 gauge and 30-gauge blunt needles. Buffered ACSF was constantly applied throughout the aspiration to prevent tissue desiccation. The aspiration ceased after full termination of bleeding, at which point a GRIN lens (1mm diameter, 4mm length, Inscopix 1050-002176) was stereotaxically lowered to the targeted implant site (-2.0 mm dorsoventral from skull surface relative to bregma). Cyanoacrylate glue was used to affix the lens to the skull. Then, dental cement sealed and covered the exposed skull, and Kwik-Sil covered the exposed GRIN lens. Carprofen (5 mg/kg) and dexamethasone (0.2mg/kg) were administered during surgery and for 7d after surgery together with amoxicillin (0.25 mg/mL) in the drinking water. 2 weeks after implantation, animals were anesthetized again with isoflurane (1–3%) and a Miniscope attached to an aluminum baseplate was placed on top of the GRIN lens. After searching the field of view for in-focus cells, the baseplate was cemented into place and the Miniscope was detached from the baseplate. A plastic cap was locked into the baseplate to protect the implant from debris.

Optogenetics

Animals were habituated to the patch-cord for 3 days in advance of optogenetic stimulation. A patch-cord was connected to the fiber optic cannula and animals were allowed to explore a clean cage for 5 minutes. On the testing day, optical stimulation through the fiber-optic connector was administered by delivering light through a patch-cord connected to a 473-nm laser (SLOC, BL473T8-100FC). Stimulation was delivered continuously with 2.5 mW power at the fiber tip.

Miniscope Recordings

Mice were handled and habituated to the weight of the microscope for 4 days before behavioral acquisition. On the recording day, a V4 Miniscope was secured to the baseplate with a set screw and the mice were allowed to acclimate in their home cage for 5 minutes. Imaging through the Miniscope took place throughout the entire PMA training (~30 min) and retrieval session the following day. Behavior was simultaneously recorded with a UCLA MiniCAM.

Behavior Video Recordings

Behavioral videos were acquired using one of the following 3 setups:

1) 50fps using a Chameleon3 3.2 megapixel monochrome USB camera fitted with a Sony 1/1.8 sensor (FLIR systems, CM3-U3-31S4M-CS) and a 1/1.8 lens with a 4.0-13mm variable focal length (Tamron, M118VM413IRCS). We recorded 8-bit videos with a 75% M-JPEG compression.
2) 30fps using a ELP 2.8-12mm Lens Varifocal Mini Box 1.3 megapixel USB Camera.
3) 50fps using UCLA MiniCam (5M CMOS sensor (MT9P031I12STM-DP, ON Semiconductor)

Histology

Mice were transcardially perfused with phosphate-buffered saline (PBS) followed by 4% paraformaldehyde (PFA) in PBS. Brains were dissected, post-fixed in 4% PFA for 12–24h and placed in 30% sucrose for 24–48 hours. They were then embedded in Optimum Cutting Temperature (OCT, Tissue Tek) and stored at -80°C until sectioning. 60um floating sections were collected into PBS. Sections were washed 3x10min in PBS and then blocked in 0.3%PBST containing 10% normal donkey serum (JacksonImmunoresearch, 17-000-121) for 2h. Sections were then stained with rabbit anti-RFP (Rockland 600-41-379 at 1:2000) in 0.3% PBST containing 3% donkey serum overnight at 4°C. The following day, sections were washed 3x5min in PBS, and then stained with secondary antibody (JacksonImmunoresearch Cy3 donkey anti-rabbit IgG(H+L) 711-165-152, 1:1000) in 0.3% PBST containing 5% donkey serum for 2 hours at room temperature, washed 5min with PBS, 15 min with PBS+DAPI (Thermofisher Scientific, D1306, 1:4000), and then 5 min with PBS. Sections were mounted on glass slides using FluoroMount-G (ThermoFisher, 00-4958-02) and then imaged at 10x with a Leica slide scanning microscope (VT1200S).

Manual annotation of behavior

Two-minute samples of each video recording were manually annotated by at least three individuals for freezing behavior. One-minute intervals were chosen from the beginning and end of the video recordings to capture diverse behaviors. Freezing was defined as the absence of movement except for respiration. Video frames (36,000 frames: PointGrey, 36,000 frames: PointGrey+Opto; Webcam: 21,600 frames; MiniCAM: 30,000) were analyzed by 3 independent manual raters per video type.

Statistical analyses

Statistical analyses were performed in MATLAB or GraphPad Prism.

Computer workstation specs

We trained networks in DLC and analyzed videos using two different custom-built workstations (Intel Core i9-9900K processor (8x 3.60GHz/16MB L3 Cache), 2x16GB DDR4-3000 RAM, NVIDIA GeForce RTX 2070 SUPER - 8GB GDDR6; AMD RYZEN 9 3950x processor (16x3.5GHz/64MB L3 Cache), 16GB DDR4 RAM, Gigabyte GeForce RTX 2060 SUPER 8GB WINDFORCE OC). BehaviorDEPOT can run on any personal computer and does not require a GPU.

Installation of BehaviorDEPOT

Detailed instructions on BehaviorDEPOT installation can be found on GitHub: https://github.com/DeNardoLab/BehaviorDEPOT. Briefly, after installing a recent version of MATLAB (2018+), BehaviorDEPOT can be downloaded from Github and installed with a single click as either a MATLAB application or as a standalone exe file. Updates to the application will be added to the Github repository as they come available. We welcome feedback and bug reports on the BehaviorDEPOT Github page and encourage users to watch the page to be aware of any new releases.

MiniCAM Instructions and Installation Descriptions of fabrication and use of MinCAMs can be found on GitHub: https://github.com/Aharoni-Lab/MiniCAM

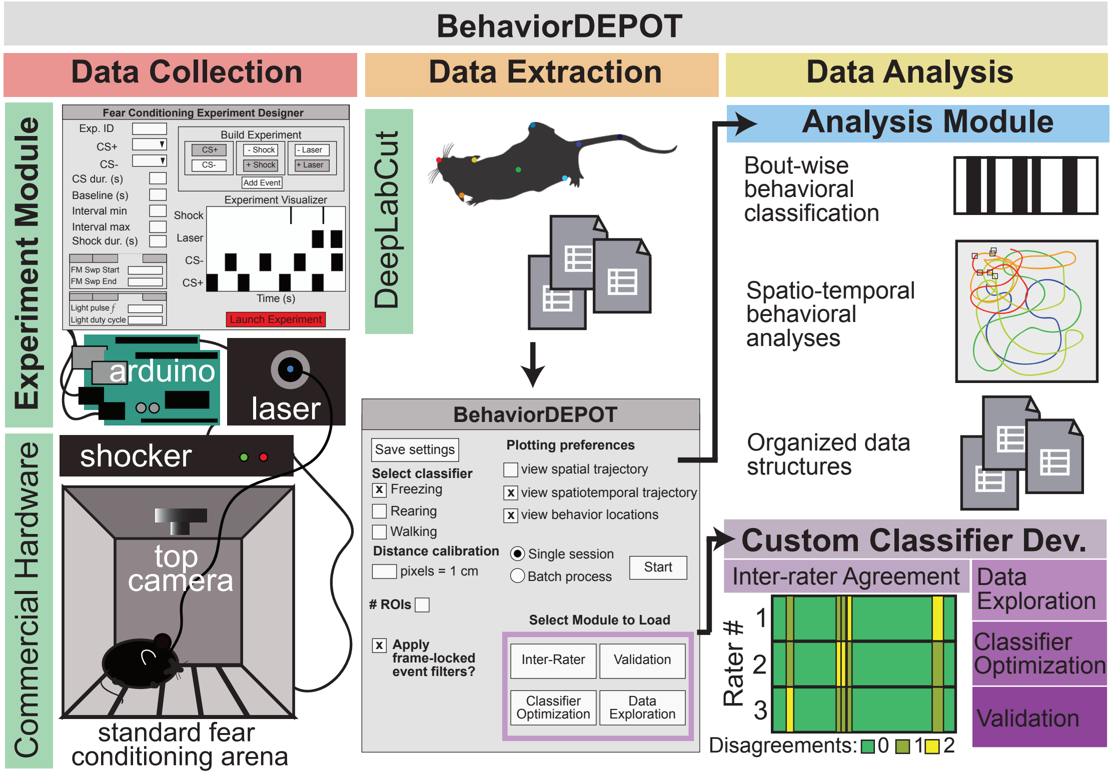


Figure 1. BehaviorDEPOT overview. The Experiment Module is a MATLAB GUI based application that allows users to design and run fear conditioning experiments. The software uses Arduinos to interface with commercially available hardware (e.g., shockers and lasers) to control stimuli. Behavior videos are acquired with webcams or machine-learning quality cameras. Video recordings are analyzed with pre-trained DeepLabCut models. The video and pose estimates are the inputs for a second MATLAB GUI that controls the Analysis Module for customizable analyses of freezing, rearing, and moving. Four additional modules help users develop custom classifiers.

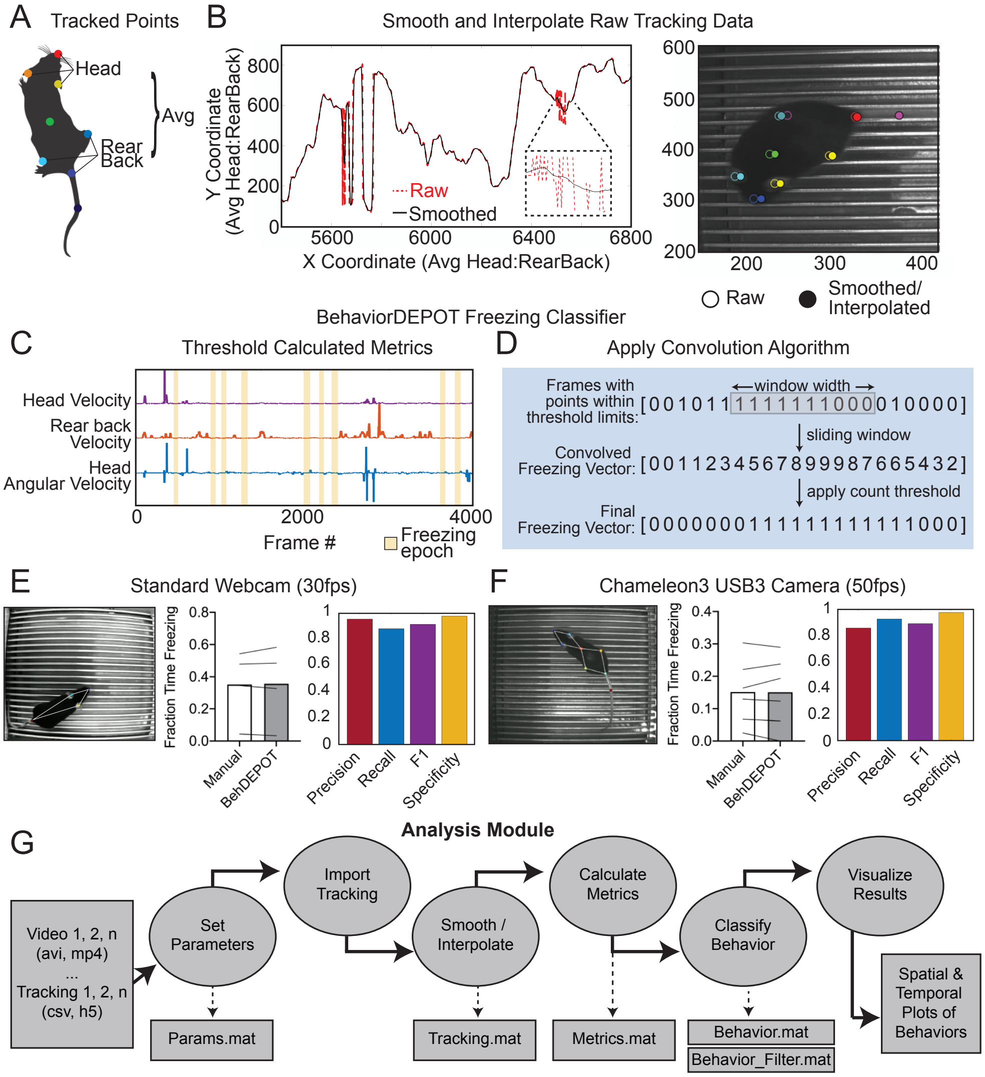


Figure 2. The Analysis Module. A. Metrics based on individual tracked points and weighted averages are calculated and stored in BehaviorDEPOT data matrices. B. Visualization of the effects of the LOWESS smoothing and interpolation algorithms for the weighted average of head and rear back (left) and for all tracked points in a representative example frame (right). C. Visualization of metrics that form the basis of the BehaviorDEPOT freezing classifier. Colored lines represent framewise calculated values for each metric. Yellow bars indicate freezing epochs. D. Visualization of the convolution algorithm employed by the BehaviorDEPOT freezing classifier. A sliding window of a specified width produces a convolved freezing vector in which each value represents the number of freezing frames visible in the window at a given frame. An adjustable count threshold converts the convolved freezing vector into the final binary freezing vector. E. Evaluation of freezing classifier performance on videos recorded at 30fps with a standard webcam. F. Evaluation of freezing classifier performance on videos recorded at 50 fps with a machine learning-quality camera. G. The Analysis Module workflow. Videos and accompanying pose tracking data are the inputs. Pose tracking and behavioral data is vectorized and saved in MATLAB structures to facilitate subsequent analyses

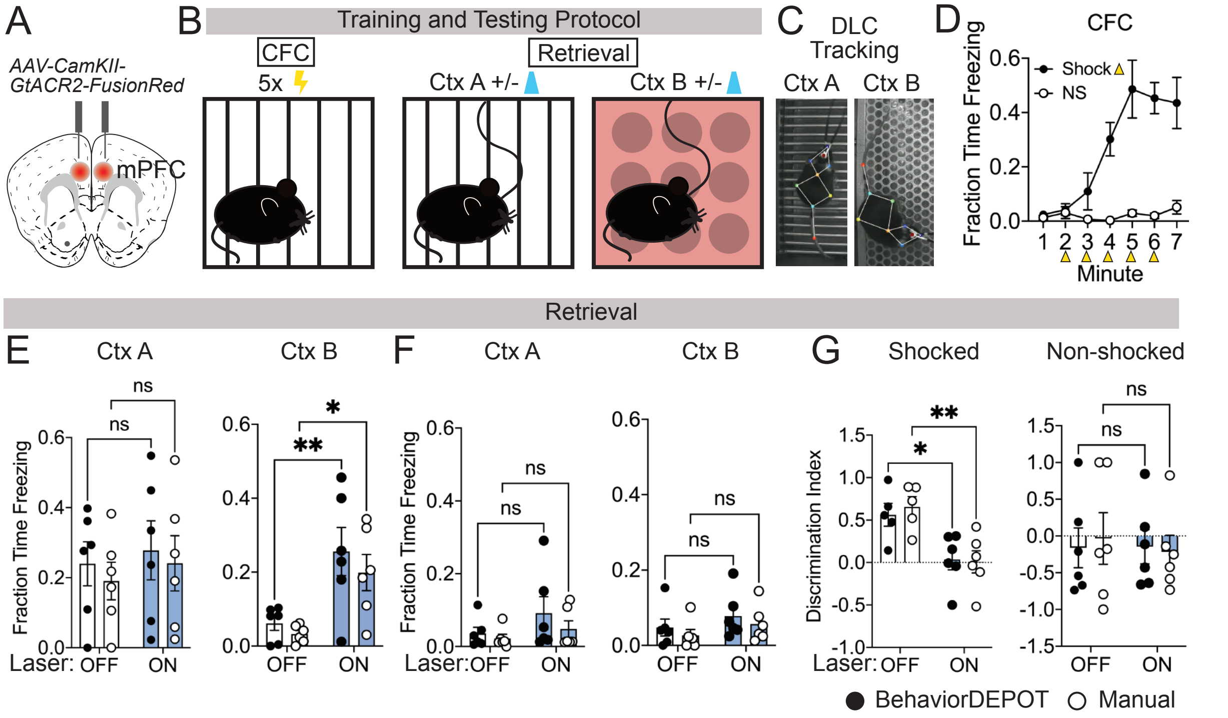


Figure 3. Use Case 1: Optogenetics. A. AAV1-CamKII-GtACR2-FusionRed was injected bilaterally into medial prefrontal cortex (mPFC). B. Behavioral protocol. Mice underwent contextual fear conditioning on day 1. On day 2, mice were returned to the conditioned context or a novel context in a counterbalanced fashion and received 2x2 min 473nm laser stimulation separated by 2 min laser off intervals. C. Example DLC tracking of mice attached to patch cords in different contexts. D. Quantification of contextual freezing during training analyzed with BehaviorDEPOT. E–F Comparing human annotations to BehaviorDEPOT freezing classifier. E. Shocked mice: freezing in context A (left) and context B (right) with and without mPFC silencing (CtxA: $F_{laser}(1,10)=0.42$, $P=0.53$; $F_{rater}(1,10)=0.35$, $P=0.57$; CtxB: $F_{laser}(1,10)=26.51$, $P=0.0004$; $F_{rater}(1,10)=0.08$, $P=0.78$;Two-way repeated measures ANOVA). F. Non-shocked controls: freezing in context A (left) and context B (right) with and without mPFC silencing ($F_{\text{laser}}(1,10)=3.60$, $P=0.09$; $F_{\text{rater}}(1,10)=0.79$, $P=0.39$; Two-way repeated measures ANOVA). G. Discrimination index = (FreezeA - FreezeB) / (FreezeA + FreezeB) for shocked mice ($F_{\text{laser}}(1,10)=17.54$, $P=0.002$; $F_{\text{rater}}(1,8)=0.09$, $P=0.77$; Mixed-effects analysis) and non-shocked controls ($F_{\text{laser}}(1,10)=0.07$, $P=0.80$; $F_{\text{rater}}(1,8)=0.02$, $P=0.90$; Two-way ANOVA).

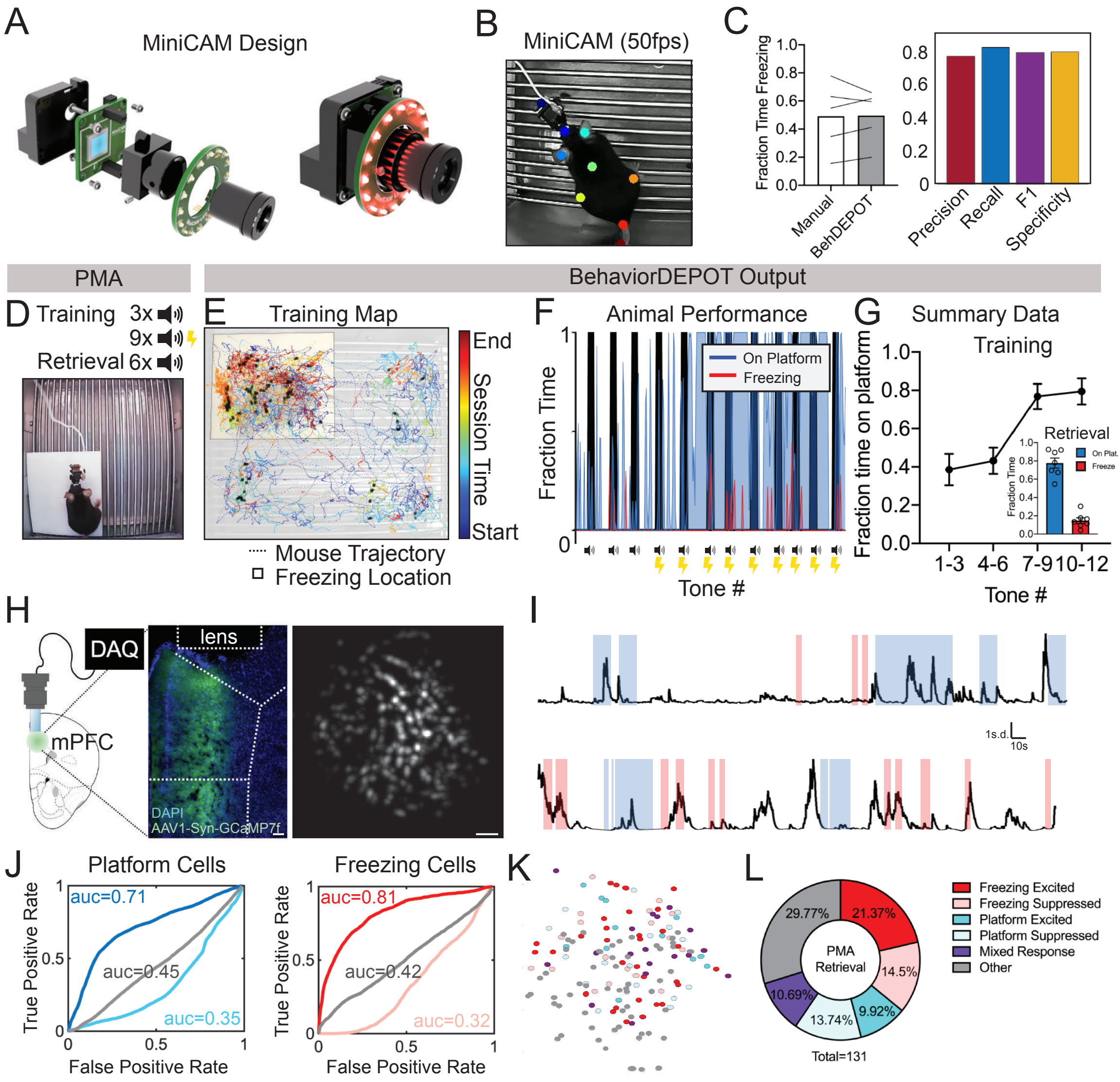


Figure 4. Use Case 2: Mice wearing Miniscopes. Design for MiniCAM, an open source camera designed to interface with Miniscopes and pose tracking. B. Still frame from MiniCAM recording of mouse wearing a V4 Minscope. DLC tracked points are labeled with rainbow dots. C. Performance of freezing classifier on videos of mouse wearing Miniscope recorded with MiniCAM. D. Task design. E. Sample BehaviorDEPOT output. Map displays animal position over time as well as freezing locations (black squares). F. Temporal alignment of time on the platform (blue), time freezing (black), and tones. G. Summary data for training and retrieval. H. GCaMP7-expressing mPFC neurons imaged through a V4 Miniscope. I. Receiver operating characteristic (ROC) curves that were calculated for platform-modulated cells (excited cell: auc-0.71; suppressed cell: auc=0.35, unmodulated cell: auc=0.45) and freezing-modulated cells (excited cell: auc=0.81; suppressed cell: auc=0.32; unmodulated cell: auc=0.42). J. Example Ca^{2+} traces from platform (top) and tone (bottom) modulated cells during time on the platform (blue) or time freezing (pink). K. Example field of view showing locations of freezing- and platform-modulated mPFC neurons. L. Proportion of modulated cells of each functional type from an individual mouse. Scale bars, 100um.

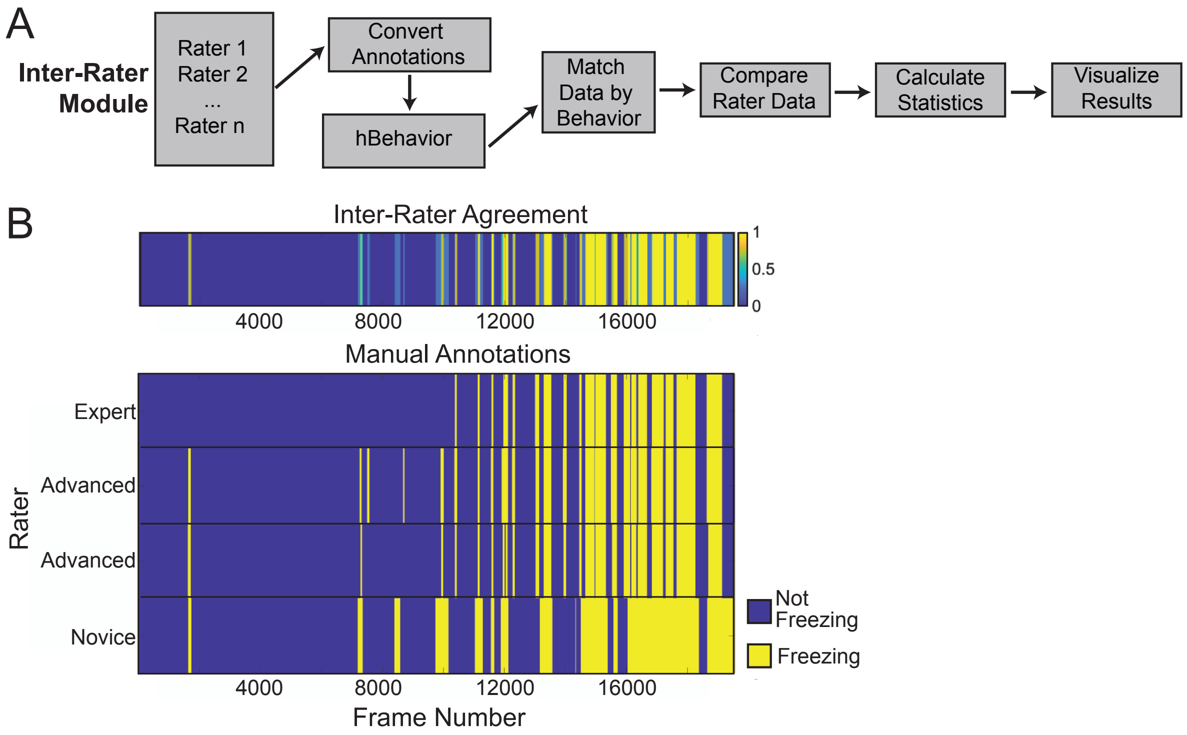


Figure 5. Inter-Rater Module. A. The Inter-Rater module workflow. B. The Inter-Rater module produces visualizations of framewise agreement levels (top) based on manual annotations from multiple human raters (bottom).

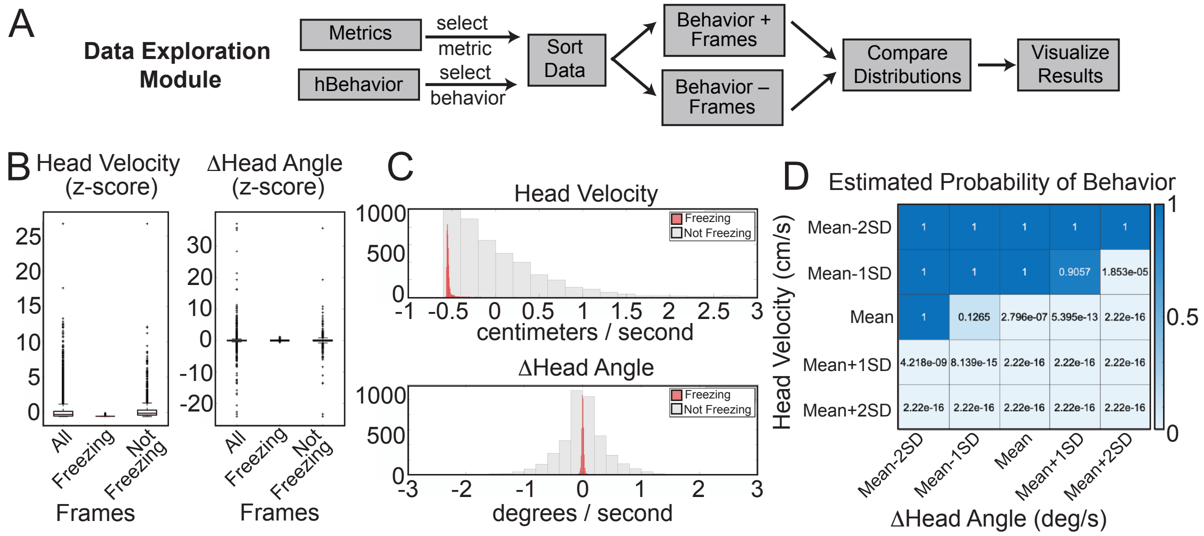


Figure 6. The Data Exploration Module. A. The Data Exploration Module takes in tracked metrics from the Analysis module and the human annotations of behavior. It sorts the data in a framewise manner, separating frames containing the behavior of interest from those without and then visualizes and compares the distribution of values for a metric of interest. B. Distributions of Z-scored values for head velocity (left) and change in head angle (right) are distinct for freezing vs. not freezing frames. C. Histograms showing distribution of values for head velocity (top) and change in head angle (bottom) for freezing (red) vs. not-freezing (grey) frames. D. A generalized linear model (GLM) computes the predictive power of given metrics for frames containing the behavior of interest.

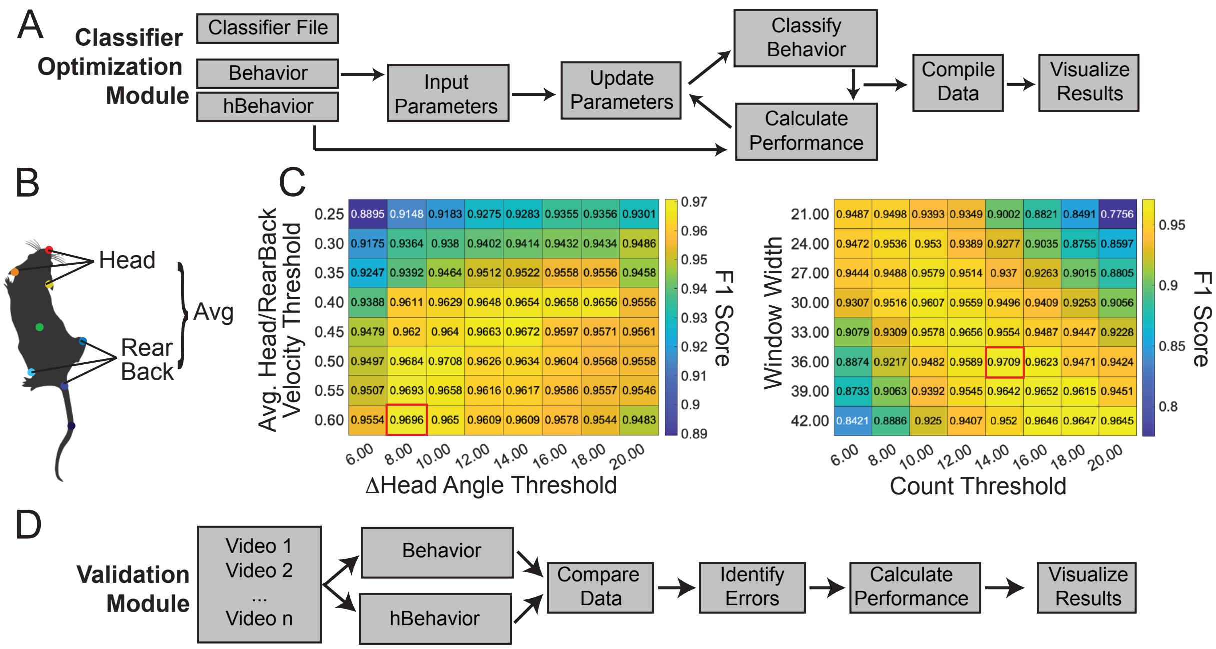
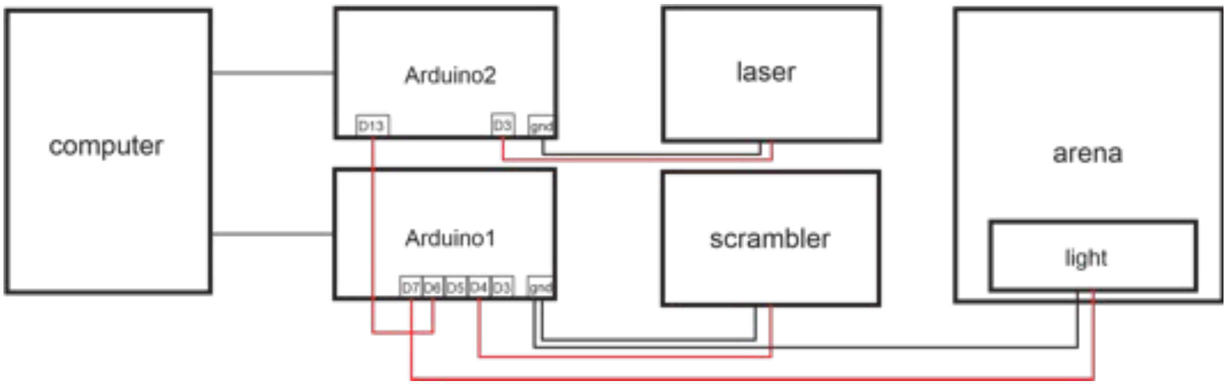
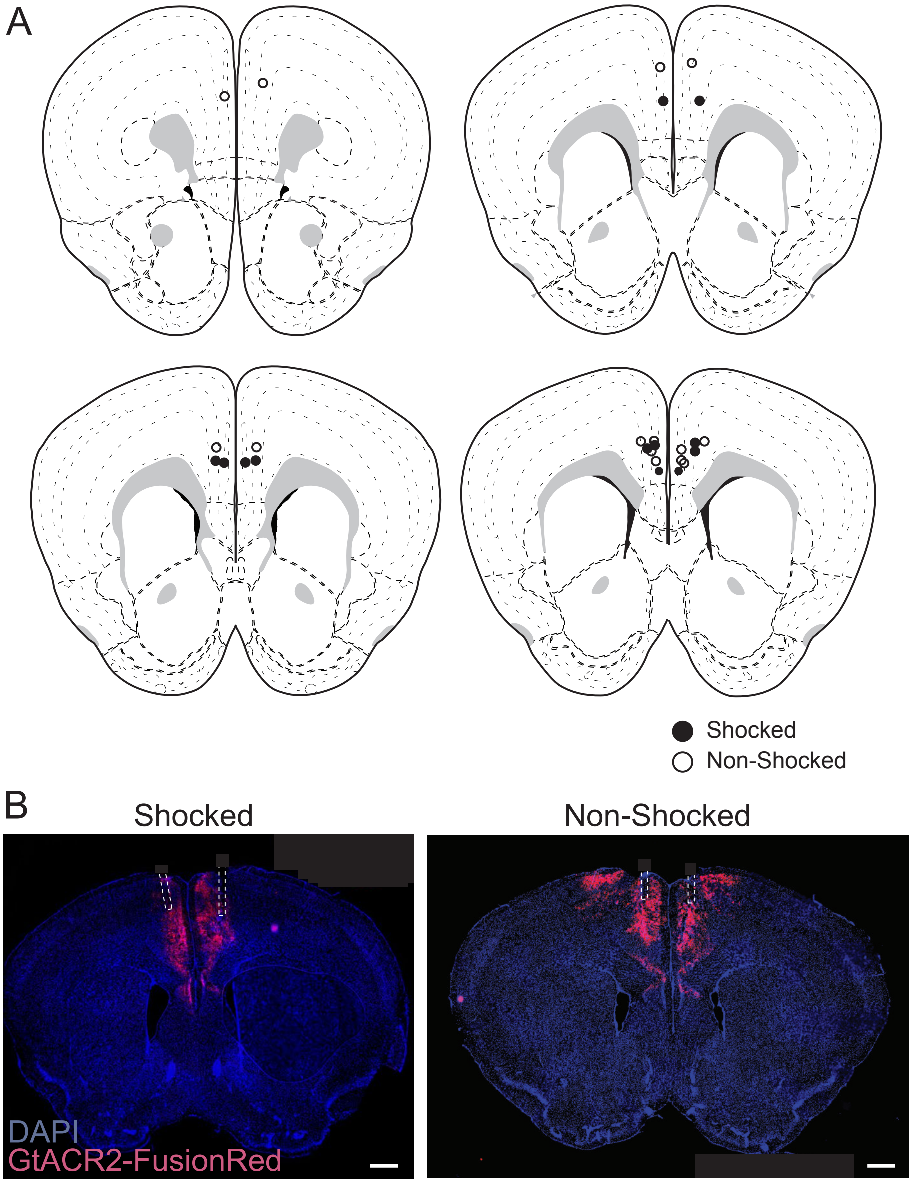


Figure 7. Classifier Optimization and Validation Modules. A. Classifier Optimization Module workflow. This module sweeps through a range of thresholds for statistics calculated based on tracked points and then compares the resulting behavioral classification to human annotations. B. We optimized our freezing classifier using weighted averages of the ears and nose (head), and the hips and midback (rear back). C. The Classifier Optimization Module iteratively sweeps through a range of thresholds for different metrics and reports F1 scores. The module first swept through a range of values for head/rear back velocity and change in head angle. The highest F1 score (red box) was selected and then a subsequent sweep through two additional value ranges (for window width and count threshold from the smoothing algorithm) produced an even higher F1 score (red box). D. Once the user has identified candidate threshold values, the Validation Module reports on recall, precision, F1, and specificity scores.



Supplemental Figure 1. Example arrangement of Arduino interface between computer and fear conditioning and optogenetics hardware. The Experiment Module controls two Arduinos that control delivery of the scrambled shocker, and a light (for use as a conditioned cue), and the laser for optogenetics, respectively. MATLAB software triggers the conditioned tone.



Supplemental Figure 2. A. Optic fiber cannula placements for experiment described in Figure 3. B. stGtACR2 -FusionRed expression and bilateral fiber placement for representative shocked and non-shocked mice. Scale bar, 500um.