communications biology

Article

https://doi.org/10.1038/s42003-024-07345-5

PreyTouch: a touchscreen-based closed-loop system for studying predator-prey interactions

Check for updates

Regev Eyal[1,2], Nitzan Albeck[1,2] & Mark Shein-Idelson[1,2]

The ability to catch prey is crucial for survival and reproduction and is subject to strong natural selection across predators. Prey capture demands the orchestrated activation of multiple brain regions and the interplay between sensory processing, decision-making, and motor execution. These factors, together with the ubiquity of prey capture across species makes it appealing for comparative studies across neuroscience and ecology. However, despite recent technological advances, experimental approaches for studying natural behaviors such as prey catch are lagging behind. To bridge this gap, we created PreyTouch—a novel approach for performing prey capture experiments that incorporate flexible prey control, accurate monitoring of predator touchscreen strikes and automated rewarding. Further, its real-time processing enables coupling predator movement and prey dynamics for studying predator-prey interactions. Finally, PreyTouch is optimized for automated long-term experiments featuring a web UI for remote control and monitoring. We successfully validated PreyTouch by conducting long-term prey capture experiments on the lizard *Pogona vitticeps*. This revealed the existence of prey preferences, complex prey attack patterns, and fast learning of prey dynamics. PreyTouch's unique features and the importance of studying prey capture behavior make it a valuable platform for connecting natural behavior with cognitive studies across various species and disciplines.

The ability to successfully identify and capture moving prey is essential for the survival, maturation, and reproduction in many animals. Capturing prey demands substantial time and resources for its planning and successful execution[1]. Consequently, it is a trait under intense selective pressure and plays a pivotal role in shaping evolutionary trajectories[2]. Reflecting its importance, prey capture was extensively studied across various disciplines ranging from ecology to psychology[3] and over diverse species spanning multiple animal classes[4–18].

Prey capture, in its complexity, demands an orchestrated activation of multiple brain regions[5,19], showcasing the intricate interplay between sensory processing[8], decision-making[20], and motor execution[17]. To successfully catch prey, animals need to first segment, identify, localize and sensorily track the prey in space[17]. Next, they need to decide whether to attack the prey by weighting multiple available options while taking into account possible risks in their surrounding environment[21]. Finally, they need to plan the attack[22] and the strike, requiring the execution of appropriate and precise motor commands[17]. Such multifaceted processing is bound to strongly depend on (and drive the evolution of) multiple computations within brain circuits. This, together with the abundance of prey catch behavior across animal classes, makes prey capture a prime paradigm for comparative studies, particularly in the context of the evolution of cognition and neural computations[23,24]. However, to exploit the potential in prey catch investigations, systematic behavioral experiments combined with electrophysiological measurements are needed.

In recent years, the field of animal behavior has undergone a significant transformation with the integration of advanced technologies[25–27]. These technologies enable the collection and analysis of extensive datasets, facilitating the exploration of new questions[28] previously inaccessible in classical prey capture studies[29]. A number of pioneering studies have already begun to leverage these technologies to explore some aspects of prey capture[8,10,11,30–32]. However, there remains a lack of open-source platforms for facilitating generalized and automated prey capture experiments, especially for freely moving animals. In particular, one potentially promising technology that has yet to be adopted for systematically studying prey capture in the lab is the use of touchscreens[33–38].

[1]School of Neurobiology, Biochemistry, and Biophysics, Tel Aviv University, Tel Aviv, Israel. [2]Sagol School of Neuroscience, Tel Aviv University, Tel Aviv, Israel. e-mail: sheinmark@tauex.tau.ac.il

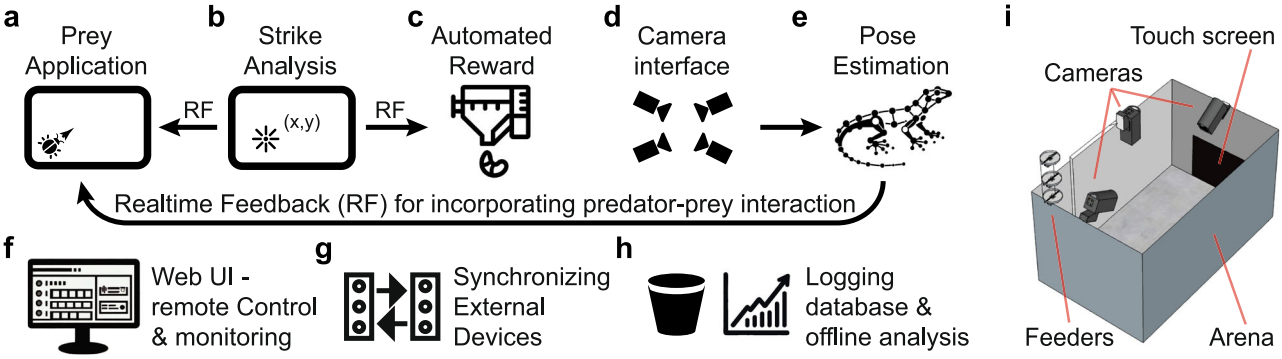

Fig. 1 | Components of the PreyTouch system. a Touchscreen prey application allowing flexible animation of moving prey. **b** Animal strikes on the screen are analyzed and sent in real time to the prey application and the reward module (**c**). **c** An automated dispenser provides rewards (including live rewards) for pre-defined touchscreen strikes. **d** A camera interface and spatial calibration module allows recording and analyzing data from multiple synchronized behavioral cameras. **e** Pose estimation and strike detection models are fed with images from the arena in real time (or offline) and provide feedback to the prey application for enabling predator-prey interactions. **f** A web-based user interface allows scheduling automated experiments as well as monitoring and manipulating experiments remotely. **g** All arena components are synchronized internally and with external devices through arduino TTL outputs. **h** Experimental information is logged in and displayed by offline modules for visualizing animal performance. **i** A scheme of the experimental arena. The arena is built with off-the-shelf aluminum profiles. A touchscreen is integrated in one of the walls in addition to cameras and a modified automated feeder that are anchored to the top frame.

To bridge this gap, we present PreyTouch—a novel touchscreen-based system for prey capture experiments. The system incorporates a touchscreen application (Fig. 1a), for presenting naturalistic and artificial prey dynamics. It supports real-time strike detection and analysis (Fig. 1b) as well as routines for feeding this information back to alter prey movement (Fig. 1a) and facilitate predator-prey interactions, or to automatically deliver rewards (including live rewards) and reinforce specific choices (Fig. 1c). PreyTouch incorporates comprehensive camera control and calibration (Fig. 1d) enabling precise frame-by-frame real-time analysis using deep models (Fig. 1e). PreyTouch includes a user-friendly web-based interface for remote control, customization, scheduling and monitoring of experiments (Fig. 1f) to support automated long-term experiments. PreyTouch ensures high precision synchronization between the system's components and with external devices (e.g., electrophysiological recordings) (Fig. 1g). Finally, all information collected by PreyTouch is logged and can be displayed with viewers for tracking of animal performance (Fig. 1h). We validated the system by successfully performing prey catch experiments in the lizard *Pogona vitticeps*.

Results

PreyTouch features and performance

Performing prey catch experiments requires a repetitive time-consuming effort. In order to streamline such experiments and scale up the number of trials, we built a low-cost ("Methods"), automated arena (Fig. 1i, see section "Methods" for details on the construction, operation, and features of the arena) and wrote a dedicated software suite named PreyTouch. The system's simple and low-cost design allowed us to easily construct multiple arenas for simultaneous recording of several animals for many consecutive days during which prey catch sessions were periodically initiated. For controlling the session times during these long experiments, we incorporated a dedicated scheduling tool ("Methods"). Each session was composed of a series of trials with a single prey item (Supplementary Fig. 1b) and a chosen prey dynamics (Supplementary Fig. 1c) presented using the prey application (Fig. 1a, "Methods", and Supplementary Video 1). We designed the application to provide animated and visually customizable stimuli mimicking live prey (Supplementary Fig. 1a, b). Further, we integrated an interface to PsychoPy to further expand the range of possible visual simulations ("Methods"). We designed a GUI for configuring all experiment parameters, for example, the quantity or frequency of sessions, the prey types and their movements, the rewards or the light/dark cycles ("Methods" and Supplementary Fig. 4f). Finally, we added a system for automatically delivering food rewards (including live prey[39]) upon successful performance and an alert system for notifying the user in case rewards needed to be replenished or upon system failures.

Measuring strike dynamics[40] can provide valuable information about the animal's prey-capture strategies. Additionally, monitoring the predator's location and posture throughout the experiment (before and after sessions) allows placing prey catch within a wider behavioral context. For this reason, we integrated into PreyTouch modules for video acquisition, calibration and analysis ("Methods" and Fig. 1d). We added a GUI for multi-camera calibration (based on Churuko markers) that allows converting all videos to accurate 2D coordinates within the arena (Supplementary Fig. 2 and "Methods"). In addition, we incorporated a video acquisition GUI allowing camera control (Supplementary Fig. 4d). Finally, we implemented two pose estimation models for extracting movement throughout experiments (Fig. 2a): Models for continuous real-time analysis (e.g., for triggering visual stimulation or external devices like heaters or lights) and slower offline models for increasing classification accuracy (Fig. 2a, b and "Methods").

We used these models to track the animal's head position and direction (Fig. 2a). During the task, this information is useful for tracking the strike dynamics and its relation to prey motion. Figure 2d shows the strike dynamics measured for the lizard *P. vitticeps* and the frog *P. bedriagae*. While in both species, strikes were characterized by acceleration during approach and deceleration before hitting the target, differences between species were apparent (Fig. 2c). Examining head movement trajectories revealed that lizards approached the screen also when sessions were not active and no prey items were displayed (Fig. 2b). Integrating these movements over longer time periods revealed a generally higher occupancy near the screen suggesting that prey catch affected the lizard's behaviors outside the task (Fig. 2b).

Predator-prey interactions are often a bi-directional processes during which both prey and predator modify their behavior in response to the other's actions. Facilitating such interactions under laboratory experimental settings requires a closed loop system. We therefore incorporated an efficient multiprocessing architecture for executing real-time visual models designed for strike detection. We used this functionality to simulate prey jumps in response to an attack by the predator[30]. One way to achieve this is to identify the position of the animal (Fig. 2d—blue box) and move the prey when the animal is proximal to the screen. However, such proximity does not necessarily indicate a strike attempt. An alternative approach is to identify a distinct pre-strike marker exhibited by the predator. For instance, some predators, like *P. vitticeps* extend their tongues just before launching an attack (Fig. 3a)[7,40]. By training a model ("Methods") to discriminate between pre-strike lizard images with extended tongues (Fig. 3a) and images

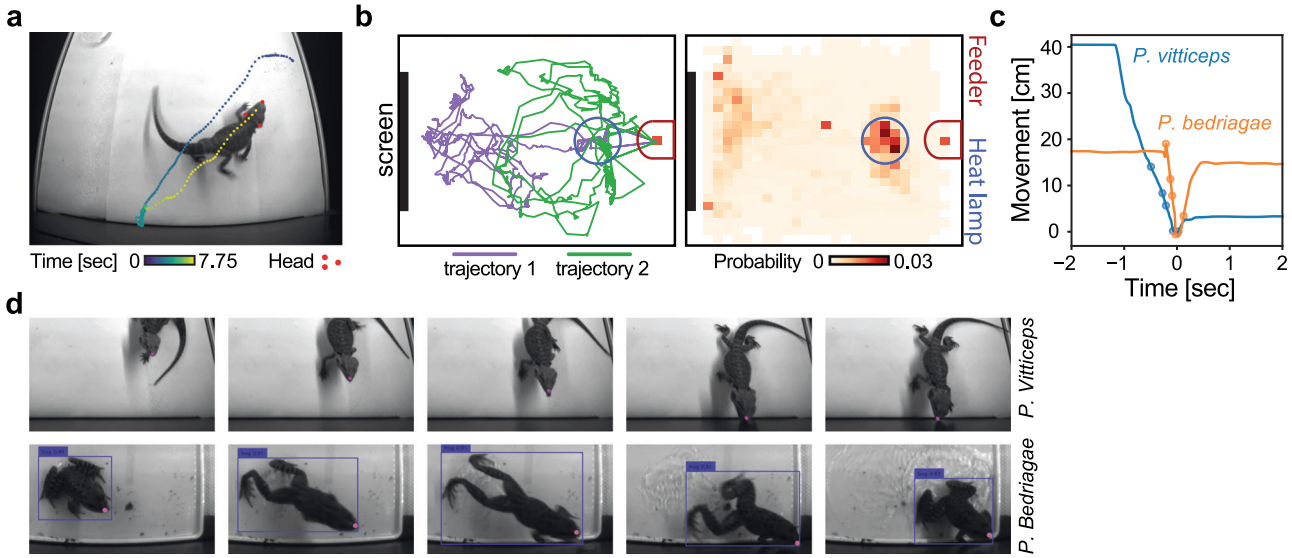

Fig. 2 | PreyTouch integrates pose estimation for measuring strike dynamics and head and body movement. a Top view of a lizard after a strike on its way to receive a reward. Head triangle (marked by red dots) is estimated using DeepLabCut and enables tracking of head movement. Color dots depict the position of the jaw tip as a function of time (color coded). **b** Long-term position tracking of a lizard in the arena. (Left) Two trajectories (green and purple) acquired during two 30 min epochs. (Right) The occupancy probability distribution (color coded) of the lizard's position accumulated over 167 h of recording during one month of experiment in one animal. The feeder and heat lamp locations are marked by red and blue lines, respectively. **c** Strike dynamics of the animal's jaw tip (pink dot in **d**) along the axis perpendicular to the screen. The dots correspond to the time of frames in (**d**). **d** A series of snapshots during prey strike for a lizard (*P. vitticeps*, top) and a frog (*P. bedriagae*, bottom). Pink dots depict jaw tips (obtained using a pre-trained DeepLabCut model) and blue rectangles depict a bounding box (obtained in real time using a Yolo5 model).

with retracted tongues (Fig. 3b), we created a simple yet effective real-time classifier for strike attempts. We achieved a low rate of false positives and false negatives (Fig. 3c) as well as AUC (area under curve) close to 1 (Supplementary Fig. 3) when running the classifier on a test set.

We used this classifier for a feedback experiment in which a prey moved along a horizontal trajectory on the screen (Supplementary Fig. 1c) and jumped up every time the lizard attempted to strike it (Supplementary Video 2). An example of the lizard's approach towards the screen, followed by two strike attempts is observed in Fig. 3d. PreyTouch successfully responded with a prey jump before the lizard hit the screen, as evident from the strike-triggered histogram of prey jump times which occurred ~200 ms before strike time (Fig. 3e). To estimate the latency in closed-loop experiments, we surrounded our processing pipeline (including image acquisition and processing, tongue detection and visual feedback) between two externally measurable events: a trigger sent to turn on a LED in the arena and the detection of visual stimulation on the arena touchscreen triggered by this LED's onset ("Methods"). The overall latency of our real-time feedback procedure was 86.8 ± 27.5 ms (mean ± s.d., Fig. 3f). This latency is well below the delay between strike detection and the lizard's strike time (Fig. 3e) and is thus adequate for manipulating prey movement in real-time. The sub-components of the measured delay are detailed in Fig. 3g.

Prey catch preference and learning in *Pogona vitticeps*

To demonstrate the performance of PreyTouch we performed a series of prey catch experiments. We focused our analysis on two prey movement patterns: linear horizontal movement and circular movement (Fig. 4a). We first examined the engagement of the lizard with the artificial prey stimulus. Following prey onset, lizards sharply moved their head towards the prey (Fig. 4c) indicating prey perception. This movement was followed by a quick approach towards the screen (Fig. 4c). Interestingly, the onset and offset of prey movement were coupled with dramatic changes in local field potentials and spiking activity recorded from the dorsal ventricular ridge (Supplementary Fig. 5 and Supplementary Video 3), demonstrating the system's compatibility with electrophysiological recordings. To further quantify head dynamics relative to the prey, we defined the prey deviation angle (Θ) as the angle between the prey position on the screen and the head direction (Fig. 4b; 0°—the head pointing directly at the prey; positive value—left eye on the prey; negative value—right eye on the prey). Following previous reports of gaze lateralization[41,42], we compared between sessions in which a prey moved horizontally from right to left (and therefore appeared first with higher probability on the right eye) and vice versa. We found similar movement characteristics in both cases, with symmetric approach patterns (Fig. 4c, d). While animals differed in the time it took them to turn their heads towards the prey and approach it (Fig. 4e), we didn't find any evidence for consistent differences in head movement as a function of prey directions which would have indicated visual lateralization. To verify this, we averaged the deviation angles ($\overline{\Theta}$) during the first second following prey appearance in each trial and tested if they significantly and consistently deviated from 0° to a particular direction ($\overline{\Theta} > 0°$ indicating left-eye lateralization and <0° indicating right-eye lateralization). We found that for two out of five tested animals, $\overline{\Theta}$ did not significantly ($p < 0.05$) differ from 0° (1-sample *t*-test; $t = -2.33$, $p = 0.21$; $t = 5.73$, $p = 0.06$), and for others, the lateralization patterns were not consistent towards a specific eye (1-sample *t*-test; $t = -56.9$, $p < 0.001$; $t = 19.84$, $p < 0.001$; $t = 11.68$, $p = 0.02$).

We next examined head dynamics during circular movement. In these trials, the prey stayed on the screen for longer periods. We observed that the lizards moved their head (quantified by the head angle, α, Fig. 4b) to follow the prey for several cycles before striking (Fig. 4f). This movement resulted in positive correlation values between the prey position and head movement (Fig. 4g—top). Such correlations were consistent across animals (Fig. 4g—bottom). Interestingly, although lizards observed and tracked the prey during all phases of the prey's circular motion, they only struck the prey when it reached the bottom of the screen (prey rotation angle of 270° ± 60°, Fig. 4h).

To study the lizard's prey catch accuracy, we analyzed the positions of strikes relative to the prey. Strike positions were registered on the screen in real time and successful strikes (defined by falling within a predefined radius surrounding the prey center; blue dots in Fig. 5a—top) were rewarded with a live mealworm delivered by the automated feeder. The strike position was asymmetric with respect to the prey direction of movement with most

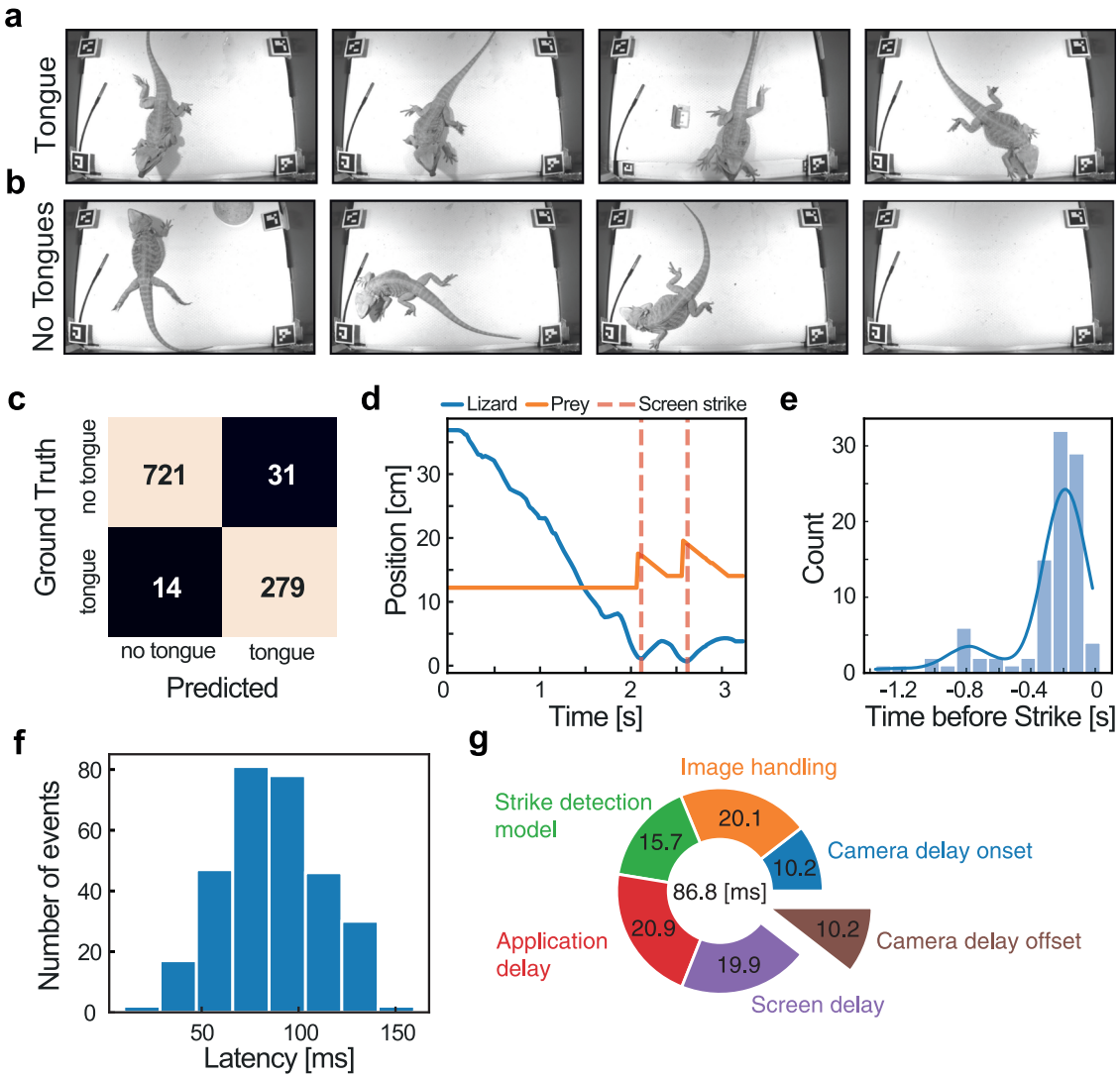

Fig. 3 | Real-time strike detection and visual feedback for bi-directional predator-prey interactions. a, **b** Sample frames used to train a deep learning model to identify the initiation of lizard strikes. The dataset included two categories: **a** frames capturing a visible tongue just before a strike begins, and (**b**) frames without visible tongues. **c** Confusion matrix for the model's performance on the test set showing a very low number of false positives and false negatives. **d** Example of strike detection and visual real time feedback during capture of a horizontally moving prey. Concomitant with the lizard's approach (blue trace marking jaw tip position on the axis perpendicular to the screen), the detection of tongue protrusions triggers prey jumps (orange curve marking the movement of the prey) occurring just before the lizard strikes the screen (dashed line). **e** A histogram of the strike detection times relative to the screen strike time ($t = 0$). Blue line marks the kernel density estimation for the distribution. The large majority of strike detections occur between 0 and 400 ms prior to the strike. **f** The overall closed loop latency distribution ("Methods") over 500 feedback events (mean ± s.d. = 86.8 ± 27.5 ms). **g** Breakdown of the overall latency to software and hardware components.

strikes located in the rear part of the prey (Fig. 5a—top). This strike pattern was consistent across lizards (Fig. 5a—bottom). Lizards managed to successfully strike the prey in a large fraction of the strikes but success rates were not homogenous across prey movement types (Fig. 5b). Lizards were more successful in striking horizontally moving prey and less successful in striking circularly moving prey (Fig. 5b) indicating a difference in task difficulty.

To understand if strike performance could be improved, we examined the miss distance (Euclidean distance between the strike position and the prey position) for each registered strike and tracked it over time. We found that lizards significantly decreased their miss distance as virtual-hunting progressed (Fig. 5c) as quantified by negative correlations of the miss distance over consecutive strikes. This decrease in miss distance was apparent also when inspecting individual animals separately (Supplementary Fig. 6a). However, the rate of decrease differed across animals, with some showing steps in their improvement along the experiment. Since some trials contained more than one strike, we averaged the miss distance over trials, revealing a similar trend of improvement in catching the circularly moving prey (Supplementary Fig. 6b). Surprisingly, we did not find a significant improvement in the miss distance during horizontal movement trials across animals ($n = 5$ animals; $r =$ correlation coefficient; $r = -0.16$, $p = 0.16$; $r = -0.33$, $p = 0.026$; $r = -0.14$, $p = 0.195$; $r = -0.18$, $p = 0.094$; $r = -0.06$; $p = 0.591$).

Finally, we examined if lizards had a preference for specific prey items. To do so, we presented prey items differing in shape and color (Fig. 5d). To quantify relative preference of different items we calculated the number of strike attempts per trial and normalized over prey items. We found that the preferred prey item across animals was the mealworm and the second most preferred item was the cockroach (Fig. 5d). Interestingly, this choice corresponded to the live feed lizards received in their regular diet. Mealworms (*T. molitor* or small *Z. morio*) were provided both in the home cage (located in the animal house where lizards were kept before the experiment) and as a reward during the experiment, and cockroaches (*N. cinerea*) were provided in the home cage. This suggests that lizards can differentiate between prey items and associate between live prey and prey presented on the screen.

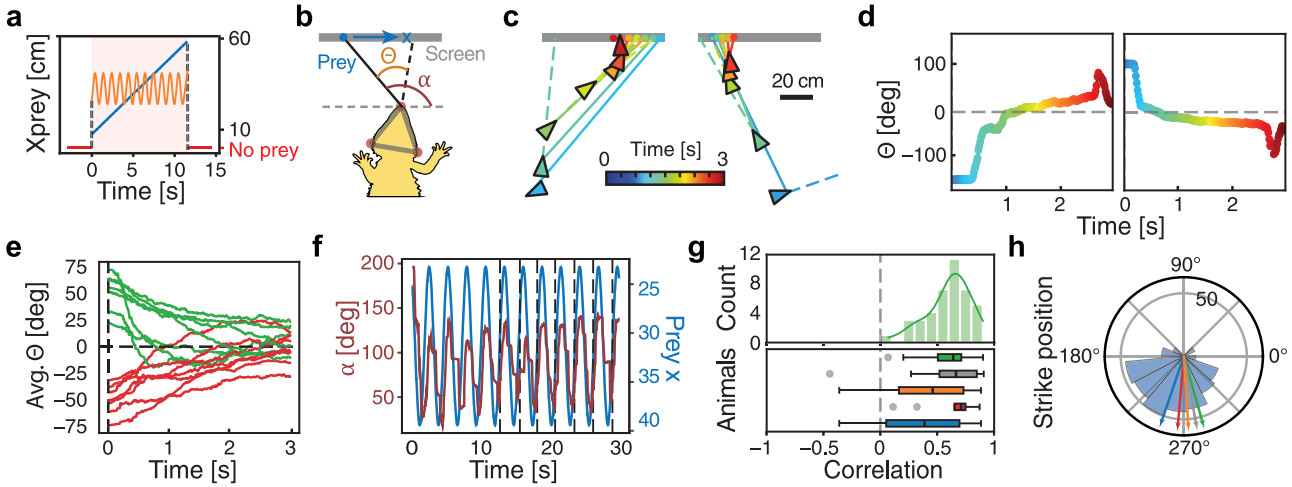

Fig. 4 | Lizards exhibit prey following. **a** Dynamics of the prey's horizontal position on the screen in horizontal (blue line) and circular (orange line) movement types. **b** A schematic illustration of a lizard during prey catch. Head position is marked by a triangle (shaded gray) whose direction (α) is determined by three head points (red dots, see also Fig. 2a). The prey is marked by a blue dot. Θ (prey deviation angle) marks the angle between the head direction (black dotted line) and prey direction (black solid line). **c** Head and prey dynamics. Head position (triangles), prey position (filled dots), head direction (dashed color lines), and head-to-prey direction (solid lines) for single frames (every 20th frame is shown) color coded for time ($t = 0$ prey appearance) during a single trial of prey catch. Two trials with prey movement to the left (left) and to the right (right) are shown. Note change in head direction following the appearance of the prey on the screen and the consequent movement towards it. **d** The deviation angle (Θ) dynamics for the trials in (**c**). Same color code as in (**c**). **e** Average deviation angle (Θ) dynamics after prey appearance ($t = 0$) for trials with prey moving horizontally from right to left (red) and from left to right (green). Lizards ($n = 5$) converge towards binocular gaze on the prey (Θ = ~0°) within a few seconds of prey onset. **f** Head angle, α (red line), and prey's horizontal position (blue line) as a function of time in circular trials. Black vertical-dotted lines indicate strike attempts. Notice locking of the head to prey motion. **g** Distributions of correlation values between the head angle and the prey's horizontal position (as in **f**) during circularly moving prey trials in one animal (top) and all animals (bottom; $n = 5$). Box plots indicating the 25th–75th percentile (middle line marks the median). Whiskers indicate the 0th and 100th percentile and outliers are marked as gray dots. **h** Distribution of prey position (angles on the circle) at strike times during circular movement sessions for one animal (blue). The average strike angle in different animals is marked by arrows. Plots (**g**, **h**) show statistics from 5 lizards with consistent color-coded identity.

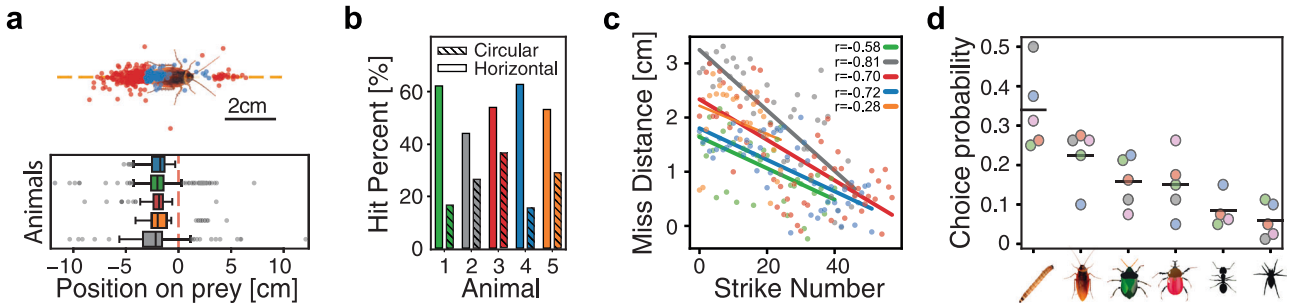

Fig. 5 | Prey learning and prey preferences during prey catch. **a**-Top Scatter plot of screen strikes recorded from a single lizard (cockroach prey). Misses (miss distance > prey body radius, 1.85 cm; $n = 316$; red) and successful rewarded hits ($n = 129$; blue) are shown. **a**-Bottom Strike hit statistics during circular movement trials for five lizards (box plots as in Fig. 4g). **b** Percentage of successful strikes for horizontal and circular movement trials across animals. Notice higher success rates for horizontal movement. **c** Dynamics of miss distance for consecutive strikes in circular trials for all animals. Improvement in the task is quantified by significant negative correlation (legend) (r correlation coefficient, p regression model p-value; $r = -0.58$, $p < 0.001$; $r = -0.81$, $p < 0.001$; $r = -0.7$, $p < 0.001$; $r = -0.72$, $p < 0.001$; $r = -0.28$, $p = 0.182$). **d** Normalized (across prey types) prey choice probabilities (fraction of strikes per number of presented trials) for different lizards. Black line marks the mean value over animals. Plots (**a**–**d**) show statistics from five lizards with consistent color coded identity (as in Fig. 4g, h).

Discussion

In this manuscript we introduce a novel experimental toolkit for conducting prey capture experiments using a touchscreen. PreyTouch is constructed from off-the-shelf low-cost components, increasing its accessibility to a variety of laboratories. The open-source nature of PreyTouch and its integration with PsychoPy empowers experimenters with easy customization for adapting the system according to specific requirements and preferred species. These properties together with its real-time processing, the long experiments scheduler and its compatibility with electrophysiological recordings (Supplementary Fig. 5) make PreyTouch a valuable tool for comparative studies across species.

PreyTouch was built for studying animal models capable of catching prey by touching a screen with their tongue (e.g., many reptiles[34] and amphibians[38] as well as birds[36,43,44]). However, PreyTouch can be potentially utilized across a much wider range of species exhibiting prey catching behaviors. Mammals were shown to interact with touchscreens[33,45,46] and can be trained to catch moving targets on a screen[35]. Our touchscreen-based prey catch approach diverges from previous research utilizing live prey[13,31,40]. The main advantage of using live prey is its realistic appearance which more likely elicits natural predator-prey interactions. Additionally, in the wild, prey capture dynamics mirror the animal's ecology and are not shaped by the incentivized learning in lab settings. However, these advantages come at the price of limited control. For example, trial to trial variability in live prey movements may change the animal's capture strategies thus limiting reproducibility. Further, it is difficult to accurately manipulate a live prey's appearance, position and dynamics. This ability to pre-determine prey

parameters is important for using prey catch paradigms to systematically study sensory processing and movement-dependent neural computations[47]. Finally, when using live prey, the number of prey catch trials is limited by the satiation of the animal. An alternative to live prey is the usage of robots mimicking prey items[10,30]. However, such robots require substantial design efforts, complex arenas and are hard to generalize to different species.

In contrast to using live prey or robots, PreyTouch allows full control over prey properties and easy placement of prey backgrounds and visual context (Fig. 1 and Supplementary Fig. 1), thus offering a good tradeoff between realism and control. In addition, the low latency real-time feedback integrated in PreyTouch exemplifies how prey-predator interactions can be integrated in experimental designs (Fig. 3). Finally, the integration of the animal's pose (extracted from recorded video) with the touchscreen strike logs offers an accurate quantification of the fine structure of animal movements and choices[48] (Figs. 2, 4, and 5). To the best of our knowledge, no other system offers these features for prey catch experiments. Importantly, the above benefits in prey control are valuable as long as animals are motivated to try and catch virtual prey on the screen. In addition, both the pose estimation model and the strike detection model were trained to maximize performance for *P. vitticeps*, and should be retrained to adapt to other species or strike profiles ("Methods").

PreyTouch provides opportunities for performing long-term automated experiments. Behavioral experiments with reptiles tend to last longer and require more repetitions, possibly due to their slower metabolism which diminishes the effectiveness of food as a motivator, lower curiosity or diminished play behavior relative to mammals and birds[49–51]. Following previous success with automated approaches[39], we addressed these caveats by integrating the PreyTouch scheduler agent allowing automatic and remote monitoring of animal progress as well as optimizing experiment schedules. This strategy enabled the collection of large prey catch datasets with minimal effort while animals remain engaged for prolonged periods of time (Fig. 4). Further, the low-cost and simple design allows replicating the experimental setup thus further increasing experimental throughput.

The experiments we performed exemplify how PreyTouch can offer new opportunities for studying animal cognition. Specifically, it enables providing insight on animal learning (Fig. 5c), decision making (Fig. 5d), motor behavior (Figs. 2c, 4f, g, and 5a), lateralization (Fig. 4e), spatial behavior (Figs. 2b, and 4c, d), and visual perception (Fig. 5d). For many years, reptiles were considered to be slow, adamant, and with limited learning capacity due to their underperformance in cognitive tasks[52]. Using PreyTouch, we observed significant and fast learning during circular prey-catching trials (Fig. 5c). This result is consistent with the idea that achieving fast learning in reptiles strongly depends on task design[49,51] highlighting the advantages of the prey catch paradigm. In contrast to the improvement with the circular prey, no learning was observed in horizontal movement trials. This may be due to the lower difficulty of this task, as supported by the higher success rates we measured for catching horizontally moving prey (Fig. 5b). This lower difficulty may have decreased the lizard's motivation to change strategy and improve[53].

We also observed that lizards can discriminate between different types of prey, in line with reports of their complex visual processing abilities[54–56]. Interestingly, we found that they preferred prey found in their regular diet (Fig. 5d). The ability to discriminate between prey on a screen suggests that PreyTouch can be used for studying visual perception and decision making. For example, we can ask to which degree can lizards associate specific prey items or movement types with rewards. Lizards also performed strikes on novel prey that they did not encounter before (Fig. 5d) implying that lizards can balance between novel and familiar prey choices. Such a property can be important if diversifying prey appearance is required in the experimental design. Further, our experiments revealed that lizards continue to strike prey on the screen despite receiving the reward in a different location on the opposite side of the arena (Figs. 2b and 5). This suggests that they can associate between the strikes and the rewards which open the door to performing visual discrimination experiments with differential rewarding. This association may have been the reason for the consistently robust prey catch behavior that allowed collecting many trials (Fig. 5).

PreyTouch can be used for studying motor behavior during prey catch. The large number of strikes we acquired and the pose estimation algorithms we used allow calculating strike profiles (Fig. 2c, d). Such analysis can be used to compare strike strategies across species and to study if different prey items or dynamics lead to different strike trajectories. Additionally, spatial analysis of approach trajectories and head movement patterns during prey catch can deepen our understanding of spatial cognition (Fig. 2b) and to link perception with motor behavior (Fig. 4c–e). For example, a report based on manually annotating the visual hemifield of the prey during lizard prey catch[41] indicated that the right eye mediates predatory activities. When automatically examining head movements and quantitatively calculating the prey's hemifield (Θ sign, Fig. 4b) over multiple trials and animals, we found no consistent lateralization (Fig. 4e). The robust quantification provided by PreyTouch allows a rich and accurate characterization of lateralization behavior and opens the door to robust comparative studies of lateralization across species.

Tracking head movements and strike choices allowed revealing prey catch strategies. While catching circularly moving prey, lizards continuously followed the prey with their heads (Fig. 4f, g). Interestingly, despite this continuous following, lizards stroke mostly at a specifically strategic point (the lower part of the screen) (Fig. 4h). Thus, lizards mostly refrained from trying to attack prey at points where it was harder to reach. Such a strategy hints that lizards possess an "understanding" of their prey's dynamics and may thus possess predictive behaviors[57]. Our system enables systematic examination of such behaviors by manipulating prey movement statistics. By measuring strike dynamics and performance one can ask if lizards can learn prey movement profiles and use them for predicting future prey position and for modifying their strikes accordingly. Specifically, one could ask if lizards use information about perceived prey velocity to modify their strike speed accordingly[13]. Using PreyTouch, such questions could be investigated across animal classes (Fig. 2d). The flexibility in monitoring and manipulating prey-catch experiments, combined with the ubiquity of prey catching behavior, makes PreyTouch a valuable platform for comparative studies of animal cognition across species, offering valuable insights into brain evolution[58–60].

Methods

Hardware

The experimental arena was constructed from off-the-shelf low-cost components (Fig. 1i and Table 1). These components include an arena cage, a touchscreen, cameras, a server, and arena peripherals and are described in detail below.

Arena Cage. The arena is built from a frame of connected aluminum profiles with dimensions of 70 × 100 × 45 cm (W × L × H) (Fig. 1i). Arena walls and floor were made from 3 mm thick aluminum composite panels[39]. The choice of conductive wall is optimized for experiments with electrophysiological measurement requiring a Faraday cage.

Touchscreen. A touchscreen (Dell P2418HT; 53 × 30 cm, 24"; Horizontal frequency: 30–83 Hz, Vertical frequency: 50–76 Hz. Resolution: 1920 × 1080, refresh rate: 60 Hz.) was integrated into one of the arena walls, and cameras as well as a modified and automated worm feeder were anchored to the arena top rail (Fig. 1i). For experiments combining electrophysiology we used a low noise touchscreen (ELO Touch Solutions AccuTouch 1790 L 17" LCD open frame) that did not induce 50 Hz artifacts when animals got close to the screen.

Cameras. Two sets of cameras were used with PreyTouch and are fully supported by the system: (1) FLIR-Firefly FFY-U3-16S2M-DL + Blackfly S BFS-U3-16S2C, (2) Allied-Vision-1800 U-158c + 1800 U-158m. Cameras were anchored to the arena's top aluminum profiles using adjustable arms (Noga Engineering & Technology, LC6100) and a custom 3D-printed connector. Video was recorded at a frame rate of 60 hz.

Table 1 | List of arena hardware components and prices

Component	Description	Price per unit (USD)
Arena frame	Aluminum profile	$12/m
Arena frame connectors	Nuts, bolts, washers, and angle brackets	$0.5
Arena walls and floor	2 × Alucobond sheet (122 × 244 cm gray matte) cut to 5 parts to fit arena walls	$98
Front and top camera	2 × Firefly FFY-U3-16S2M-DL	$598
Front and top camera lens	Boowon 6”-BW60BLF	$17
Back camera (color)	FLIR Blackfly-BFS-U3-16S2C-CS	$364
Back camera lens	Computar A4Z2812CS-MPIR, 2.8–10 mm 1/2.7”, CS mount lens	$109
Camera holder	NOGA Modular holders LC6100	$124
Arena computer	Intel Core i7-11700K CPU, 32GB DDR4 memory, NVIDIA GEFORCE RTX 3080 Ti GPU, 500GB SAMSUNG 980 M.2 NVME SSD, and a 2TB 7200 RPM HDD	$3,245
Arduino microcontrollers	Arduino Nano Every	$20
Relay board (lights and heat lamps)	Pololu Basic SPDT Relay Carrier with 5VDC Relay (Assembled)-Omron G5LE-14-DC5 SPDT	$9
Reward feeder	EVNICE EV200GW fish feeder	$51
LED Strip	4 meter, 12 V Cold-White approx. 1 A/meter	$20
Lamp holder	MULTICOMP G6-R32HKY-12 lamp holder	$3
Temperature sensor	Plastic casing water-proof DS18B20 temperature sensor	$22
Feeder motor driver	ULN2003 stepper motor driver board	$4
Led strip PSU	12 V 5 A AC-DC power supply	$13
Feeder PSU	5 V 500 mA AC-DC power supply	$5
Touchscreen	Dell P2418HT; 53 × 30 cm, 24”; horizontal frequency: 30-83 Hz, vertical frequency: 50–76 Hz. Resolution: 1920 × 1080, refresh rate: 60 Hz	$581
sum		$5331

Server. PreyTouch was installed on a linux desktop computer (ubuntu 22.04) with the following specs: CPU: Intel Core i7-11700K, memory: 32GB DDR4, GPU Nvidia GeForce RTX 3080Ti (nvidia-driver: 530.30.02, cuda: 11.7), storage disks: 500GB SAMSUNG 980 M.2 NVME SSD + 2TB 7200 RPM HDD.

Arena peripherals. All peripheral devices, including the reward dispenser, LED lights, temperature sensors, and camera triggers, are interfaced using an Arduino (Nano Every). The complete code for both the Arduino and the service facilitating communication through MQTT were sourced from the ReptiLearn project[39].

Lights. The arena was lit using an LED strip (12 V, 6500 K white LEDs, ~3.4 m long) that was attached using adhesive to profiles at the top edge of the four arena walls. This provided relatively uniform lighting across the arena, minimizing shadows. The strip was controlled using a relay module (based on an Omron G5LE-14-DC5 5VDC SPDT relay). The module's EN, VDD, and GND control ports were connected to one of the Arduino boards that sent on/off TTLs to control the DC output connected to the LED strip's power supply unit (12 V, 5 A).

Live prey dispenser. For rewarding animals with live prey, we utilized a readily available aquarium feeder (EVNICE EV200GW), which was attached to the arena's structure using its built-in clamp. The feeder was modified to decrease its response time and driven by an Arduino board[39]. Feeders were stocked with worms, each housed in individual compartments (15 in total) with sufficient food for gut-loading. To ensure the animals remained eager for the rewards, smaller worms proved more effective, leading us to choose *Tenebrio molitor* larvae. In instances requiring a larger worm supply between refills, we optimized space by vertically arranging the feeders, achieving capacities of up to 45 worms in three vertically aligned feeders. Feeders stacked vertically were positioned such that the upper feeder dispensed rewards through the release aperture of the feeder below it. The experimental module kept track of the reward inventory in each feeder to determine which unit should release a reward next.

Temperature sensors. We employed two DS18B20 digital temperature sensors, each encased in plastic, to monitor the temperature within the arena. The first sensor was affixed to the arena's rear wall to gauge ambient temperatures, while the second was positioned at the base of the wall opposite, directly beneath the heat lamp in the basking area. These sensors were linked to an Arduino board for data collection and 5 V supply.

Electrophysiology

Recording system. Recordings were performed with an Open EPyhs acquisition system (acquisition board v2.2) and Intan amplifier head stages (RHD2132-#C3314) connected using an ultra-thin SPI cable (RHD2000-#C3216). The Open EPhys GUI[62], was used for recording. Electrode recordings were referenced to an implanted chlorinated silver wire (a-m systems, cat-786000). During the recording, the weight of the Intan head stage was balanced using a pulley system.

Surgery. Twenty-four hours before surgery, analgesics (Meloxicam: 0.2 mg/kg or Carprofen 2 mg/kg) and antibiotics (Baytril: 5 mg/kg) were administered. On the day of the surgery, the animal was initially anesthetized with inhalation of Isoflurane in an induction box and later intubated and connected to a ventilation system (AWS 100, Hallowell EMC) maintaining a constant flow of 4% Isoflurane. Once deep anesthesia was verified, the lizards were placed in a stereotactic apparatus (RWD 68409). Body temperature was maintained via a heating pad attached to the stereotactic table at 35 °C. Eyes were protected by covering with ointment (Doratears). The skin covering the skull was disinfected with Povidone Iodine 10% and coated with a Lidocaine ointment (2%) for local analgesia. The skin on the skull was removed with a scalpel and residual tissue was removed with a micro curette and dissolved with 30% hydrogen peroxide. A small craniotomy was drilled and the dura and arachnoid layers covering the forebrain were removed with fine forceps,

and the pia was gently opened over the area of electrode insertion. A small incision in the brain was made with a 22 G needle lowered with a needle holder stereotaxic arm. A 32-channel flexible (4 µm thick) polyamide probe with TiN electrodes (custom fabricated by NMI, Germany) was inserted in the same location using a guide needle with manual adjustment. To target the DVR, the probe was lowered to a depth of 1.5 mm below the cortical surface. The brain was then covered with Dora-gel (Cambridge neurotech). The remaining exposed skull was covered with UV glue (Transbond™ XT 3 M) and dental cement (Coral Fix) to secure the electrode and reference wire. Four holes were drilled contralaterally to the craniotomy and used for the placement of anchoring screws. The lizard was then removed from the stereotaxic table to a recovery box where it remained ventilated with Oxygen until it self-extubated. Following the surgery, the lizard received a daily dose (injected subcutaneously) of analgesics (Meloxicam: 0.2 mg/kg SC), antibiotics (Baytril: 5 mg/kg SC), and 1 ml of Saline for 5 days. Recordings began after the lizard resumed normal behavior.

Software

PreyTouch software. The majority of the PreyTouch codebase was developed and tested using Python 3.8. To enable concurrent processing with multiple cameras, the system employs a multiprocessing architecture with a Redis store. The software leverages Flask API (v2.2.2) for execution and hosts a user-friendly management web interface (Supplementary Fig. 4) created using jQuery (v1.11.0). PreyTouch is compatible with touchscreens and has the capability to exert full control over them, facilitating the display of the prey application (as detailed in the subsequent section), and monitoring and logging screen touches. Communication within the system, such as interactions between camera processes, the API, the prey application, and peripheral components, is established through the MQTT messaging protocol. For instance, commands are transmitted to operate the feeder dispenser (reward) or to receive instructions from the server (e.g., initiation of prey trials or other events like prey jumps). Via the MQTT protocol, the application systematically communicates a comprehensive log of all behavioral events, encompassing screen touches and prey trajectories.

Prey application. A prey application (Fig. 1a) was designed to provide realistic visual stimuli mimicking live prey. To achieve this, each prey item was constructed of a sequence of still images that were repeatedly displayed to create animation (Supplementary Fig. 1a) in addition to an image displayed following a successful hit (Supplementary Fig. 1a—left). Prey items can be selected from an existing repertoire (Supplementary Fig. 1b) or provided as input for seamless customization of new prey items. In addition to the appearance of prey, its dynamics is also configurable. Movement profiles can be selected from a range of implemented profiles (Supplementary Fig. 1c) or customized by providing the (x, y) trajectory. In addition, the application allows determining parameters such as movement duration, velocity or prey size. Finally, static objects (e.g., walls, obstacles) can be added if necessary (Supplementary Fig. 1c). The prey application receives input from the touchscreen and responds to animal strikes accordingly (Supplementary Fig. 1a). The Prey application is written in Vue.js, a JavaScript framework that is well-suited for web-based (browser independent) animation with asynchronous events required for fast responsiveness. In addition, the application is based on MQTT messaging, allowing it to efficiently communicate with devices. This design scheme allows fast updating of prey appearance following a successful strike, sending commands to operate the reward dispenser or receiving commands from the server. The MQTT protocol is also used to send a full log of all behavioral events, such as screen touches and prey trajectories.

Logging and databases. The system performs logging (Fig. 1h) through two mechanisms: storing data in local files and a designated database. PreyTouch utilizes PostgreSQL as its chosen database, with communication facilitated through SQLAlchemy (v1.4.39). Additionally, the system incorporates a migration method to the database using Alembic (v1.8.1). Various data types are consistently saved to both the database and local files: (1) videos from the arena (actual videos are saved locally, while frame times are saved to both local files and DB), (2) screen touches, (3) prey trajectories on the screen, (4) experiment timings and configuration, (5) temperature in the arena, (6) off-line analysis, (7) rewards. For more information on the structure of the database please refer to Supplementary Fig. 7.

Detection models. Monitoring predator dynamics throughout the experiment is important for gaining a comprehensive understanding of prey catch behavior. For this reason, we integrated into PreyTouch modules for automated video acquisition from multiple cameras (Fig. 1d), in addition to pose estimation models for extracting movement information (e.g., DeepLabCut[25]). These models can operate in two modes: (1) In real-time mode, video frames are continuously analyzed for activity based triggering of visual stimulation or external devices like heaters or lights. (2) In offline mode, deeper and slower models are utilized (when the system is not actively running experiments) for increasing classification accuracy. Support is already integrated for a range of models, including DeepLabCut, YOLO, and any models based on the PyTorch framework. For instructions on how to train these models for other species or pre-strike behaviors, please visit the corresponding github page.

Tongue model—real-time strike detection. The tongue detection model employs ResNet18 as the backbone to embed input images from $224 \times 224 \times 3$ parameters to 512 parameters. The model, developed in PyTorch (v2.0.1), was designed as follows:

(resnet): resnet18(in_features = 150528, out_features = 512)
(fc1): Linear(in_features = 512, out_features = 120, bias = True)
(fc2): Linear(in_features = 120, out_features = 60, bias = True)
(fc3): Linear(in_features = 60, out_features = 2, bias = True)
(dropout): Dropout(p = 0.2, inplace = False)
(norm): BatchNorm1d(512, eps = 1e-05, momentum = 0.1, affine = True, track_running_stats = True)

Scheduler and Agent. In order to streamline experiments and scale up the number of trials, we incorporated a scheduling tool in PreyTouch. This tool allows performing automated experiments without user intervention. Before the beginning of the experiment, the arena is configured and parameters such as the quantity, frequency, types of prey-catching experiments, rewards and the light/dark cycles, are set. After configuration, animals can be placed in the system for extended periods of days, thus, allowing the system to accumulate valuable prey catch data. The only required human intervention is filling up the reward dispensers. Further, the system tracks the feeders' status and sends an alert to the experimenter in case the rewards run out. Alerts are also automatically sent in case of system failures. The scheduler, implemented as a Python module, initializes with the system, conducts checks, and executes scheduled tasks at their designated times. Examples of checks performed by the scheduler include monitoring the state of arena lights, initiating or concluding camera acquisition, commencing scheduled experiments, executing offline analysis, managing database migrations, and compressing videos. Configuration adjustments for the scheduler can be executed either through its module (scheduler.py) or via the configuration file. For detailed information on scheduler configuration, please consult the GitHub page. On the other hand, the agent, guided by a plan outlined in agent_config.json, autonomously launches experiments. It actively monitors the animal's state and performance, adjusting experiments accordingly. For guidance on configuring the agent, refer to the GitHub link.

Camera calibration and real-world coordinates. PreyTouch integrates a toolbox for easy camera undistortion (based on checkerboard detection) and for converting pose estimation results to 2D coordinates within

the arena (Using ChArUco markers). This tool is based on the openCV package for computer vision and its integrated camera calibration functions. The first step of the calibration process is undistortion and it requires a printed checkerboard (size: 9×6 square corners) that is captured by the camera in various orientations within the arena. The board is detected using the openCV *findChessboardCorners* function. The detected corners of the chess board are then fed to the *calibrateCamera* function which provides the undistort-homography-matrix (see Supplementary Fig. 2 for the result of applying the undistort-homography-matrix on a frame). The resulting undistort-homography matrix is saved by date and camera name, facilitating subsequent offline analyzes of frame data. PreyTouch offers a method for extracting real-world coordinates by converting frame detections from pixels to centimeters within a predefined coordinate system in the arena. To achieve this, users are instructed to print a ChArUco board matching the arena's dimensions and place it within. The detection of the ChArUco board is made by the openCV *cv2.aruco.detectMarkers* function that provides the corners of the detected markers with their marker ID. These detected points are first undistorted using the undistort-homography matrix. Since the board matches the arena dimensions it is possible to calculate the real-world locations of each marker. Using the undistorted locations of the markers and their real-world locations PreyTouch calculates the real-world-projection-matrix using the openCV *findHomography* function, and like the undistort matrix it is saved by camera and date. This procedure is conducted on a per-camera basis, provided the camera remains stationary. In the event of any camera movement, it becomes imperative to repeat this process. For more details on the process and explanations for how to run it in PreyTouch see camera calibration guide.

Integration with OpenEphys. PreyTouch was tested with the OpenEphys electrophysiology recording system. The cameras in PreyTouch are set in trigger mode, which is controlled by the PreyTouch software through a camera trigger Arduino. To ensure synchronization with OpenEphys, PreyTouch pauses the camera trigger for 8 s at the start and end of each experiment block. This pause allows one to identify these trigger interruptions in the recorded OpenEphys digital data and accurately extract the correct experiment triggers. Additionally, PreyTouch activates the IR LED for 1 s at the beginning of a block immediately after the trigger pause and at the end just before the trigger pause. This IR signal was used to confirm that the synchronization is functioning correctly by comparing video signals with trigger times.

Integration with PsychoPy. Experiments in the visual and auditory domains developed using the PsychoPy framework[63] can be easily implemented in PreyTouch. This is achieved by transferring the PsychoPy experiment files to a specific directory within PreyTouch (detailed guidelines are provided on the PreyTouch GitHub repository).

Prey catch task

Lizards were introduced into the experimental arena that included a heat lamp, a small shelter facing the screen, and a water dish. The experiments began after a 12-hour acclimation period in their new environment. The primary source of nourishment for the lizards came from the feeder dispensers (*Tenebrio molitor* larvae) awarded upon successful virtual insect captures (one worm for each successful attempt). Additionally, the lizards received greens once a week.

The experimental activities occurred daily from 08:00 to 17:00, with a series of trials conducted hourly. The experiment involved various trial types, each with specific goals that needed to be met before moving on to the next phase. The system automatically calculated these goals and scheduled the sessions.

Experimental Phases:

1. Random Low Horizontal: Insects move horizontally between holes at random speeds (2, 4, 6, 8 cm/s) and directions. The objective was to achieve 30 hits at each speed.
2. Circular Movement: Insects move in a circular path with random speeds (2, 4, 6, 8 cm/s) and directions. The aim was to secure 20 hits at each speed, with a 20% chance of receiving a reward on a miss.
3. Low Horizontal: Insects move at a constant speed of 6 cm/s horizontally, starting from right to left for 100 trials before switching directions for another 100 trials.

Analysis

Alignment of prey data and pose. The synchronization of camera clocks with the server clock is lacking, resulting in a misalignment between the extracted animal pose from the camera frames and the screen events obtained from the prey (bug) application. To address this disparity, when commencing an experiment, we initiate a query to each camera to retrieve its current time. Subsequently, we calculate the time difference in nanoseconds between the server time and the camera time, preserving this value for each camera process. This calculated difference serves the purpose of converting frame timestamps to align with the server time.

Evaluation of classification models. All the assessment of classification models, such as tongue-out detection, is conducted utilizing the scikit-learn package (v1.1.2) for metrics. Upon the completion of model training on the train-set, the test-set (observations unseen during training) is subjected to scikit-learn methods to compute the following metrics: Confusion Matrix: A pivotal table utilized in the evaluation of classification models, presenting counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions, summarizing model performance comprehensively.

ROC curve. (Receiver operating characteristic) Offering insights into the sensitivity (true positive rate) and specificity (false positive rate) trade-off across diverse decision thresholds, the ROC curve is a valuable tool for understanding model performance.

Precision-recall curve. Particularly essential in imbalanced scenarios, the precision-recall curve delineates the trade-off between precision and recall. It aids in threshold selection and provides a comprehensive view of a model's efficacy in capturing positive instances. The area under the PR curve (AUC-PR or AP) condenses the information from the entire PR graph into a single metric, with a higher AP indicating superior overall model performance, serving as a useful summary measure.

Real-time latency measurements. To measure real-time feedback latency, we modified the original processing block to incorporate precise external timestamps at its start and end points. Specifically, instead of using the tongue protrusion event to trigger the strike cascade, we activated an infrared LED light using an external trigger and added code to detect the LED's onset in the camera frame. This detection was used to trigger the strike cascade but we continued to process camera frames using the tongue detection model to incorporate its latency. The prey application was further modified to darken the entire screen instead of prey jumping or accelerating. The camera frame exhibiting darkening was then detected using an intensity threshold crossing. Accurate event timestamps were recorded in OpenEphys. Start time was marked by the digital trigger turning on the IR light. End time was marked as the digital trigger sent by the camera to OpenEphys during the acquisition of the darkening frame. Thus, the end-to-end latency of real-time strike detection was defined as the time from IR light onset to the darkening of the screen (Fig. 3f). Furthermore, using the PreyTouch software, we measured the various subdivisions of software latencies that contribute to this overall end-to-end latency using the PC clock. These included: (1) image handling – retrieving the arena frame from the camera's memory buffer and resizing it to fit the model, (2) model prediction of tongue-out detection, and (3) sending the "tongue detected" event to the screen application in the browser, triggering the prey's jump or acceleration

Table 2 | List of animals participating in the study

ID	Species	Sex	Weight [g]	Age [years]
PV 1	*Pogona Vitticeps*	M	203	2.1
PV 2	*Pogona Vitticeps*	M	185	1.4
PV 3	*Pogona Vitticeps*	F	214	2.2
PV 4	*Pogona Vitticeps*	F	201	1.4
PV 5	*Pogona Vitticeps*	M	194	1.1
BP 1	*Pelophylax Bedriagae*	F	82	1.5

(Fig. 3g). In addition to these software delays, there are two hardware delays which cannot be measured by PreyTouch. To measure the camera acquisition delay, we used OpenEphys triggers to measure the time from the IR light onset trigger to the camera frame trigger during the first frame in which increased IR illumination was detected (10.2 ± 7.1 ms). The time from sending the command to change the frame displayed on the screen (prey jump) until this frame was displayed was not measured directly but derived by subtracting all measurable latencies from the total end-to-end latency (Fig. 3g). Our calculation included two frame acquisition delays. One between the LED's onset and the time of increased intensity frame trigger, and one between the screen offset and the decreased intensity frame trigger. However, the actual end-to-end latency does not include the second frame delay. We therefore subtracted one camera delay (10.2 ± 7.1 ms) from the final latency results. These delays were calculated with a camera and a screen with a 60 Hz fresh rate. Higher refresh rates could be used if shorter delays are needed.

Statistics and reproducibility

Regression models. All regression models in the paper (Fig. 4) are calculated using the Scipy (v1.10) method of stats.lingress with 2-sided alternative hypothesis. The lingress method outputs the slope and intercept of the regression model, the Pearson correlation coefficient (r-value) and the p-value for a hypothesis test whose null hypothesis is that the slope is zero, using Wald Test with t-distribution of the test statistic.

Box plots. All the box plots in the paper were created using Seaborn (v0.13). The box represents the interquartile range (IQR), the middle bin is the median and the "whiskers" extend to points that lie within 1.5 IQRs of the lower and upper quartile. Observations falling outside this range are displayed as outliers.

Sample sizes were selected according to common practices in the field taking into account the nature of the experiments performed and the availability of the non-conventional animal species studied. Results were replicated across independent experiments with the exact number of statistical replicates noted in the figure legends and text.

Animals

Pogona vitticeps. Lizards were purchased from local dealers and housed in an animal house at I. Meier Segals Garden for Zoological Research at Tel Aviv University. Lizards were kept in a 12–12 h light (07:00–19:00) and dark cycle and a room temperature of 24 °C. Prior to entering the arena, lizards (Table 2) are fed a diet comprising mealworms (*Tenebrio molitor*), cockroaches (*Nauphoeta cinerea*), and vegetables such as carrots and parsley. Once introduced to the arena, the lizards exclusively receive mealworms from the automated reward dispenser with a petri dish containing vegetables provided at the end of the week. A water bowl is always accessible.

Pelophylax bedriagae. Frogs were caught from the wild and housed in an outdoor pond at I. Meier Segals Garden for Zoological Research at Tel Aviv University. Before introducing the frog to the arena, the top of the arena is covered with a net (preventing the frog from leaping out), and a large plate with dechlorinated water is placed inside the arena in proximity to the screen. The frog was rewarded with mealworms (*Tenebrio molitor*) from the dispenser upon successful strikes.

We have complied with all relevant ethical regulations for animal use. Ethical protocols were approved by Tel Aviv University ethical committee (Approval numbers: TAU-LS-IL-2201-112-4, 04-21-055, TAU-LS-IL-2208-155-4).

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The datasets used to generate the figures together with the code for generating them is available from GitHub (https://github.com/EvolutionaryNeuralCodingLab/PreyTouch) as well as Zenodo[61].

Code availability

All code and arena construction instructions can be accessed on GitHub (https://github.com/EvolutionaryNeuralCodingLab/PreyTouch) as well as Zenodo[61].

Received: 3 August 2024; Accepted: 2 December 2024;
Published online: 19 December 2024

References

1. Abrams, P. A. Foraging time optimization and interactions in food webs. *Am. Nat.* **124**, 80–96 (1984).
2. Stephens, D. W. & Krebs, J. R. *Foraging Theory*. Vol. 1 (Princeton University Press, 1986).
3. Stephens, D.W., Brown, J.S. & Ydenberg R.C. (eds) *Foraging: Behavior and Ecology*. (University of Chicago Press, 2007).
4. Konishi, M. How the owl tracks its prey: experiments with trained barn owls reveal how their acute sense of hearing enables them to catch prey in the dark. *Am. Sci.* **61**, 414–424 (1973).
5. Ewert, J.-P. et al. Neural modulation of visuomotor functions underlying prey-catching behaviour in anurans: perception, attention, motor performance, learning. *Comp. Biochem. Physiol. A. Mol. Integr. Physiol.* **128**, 417–460 (2001).
6. Hedenström, A. & Rosén, M. Predator versus prey: on aerial hunting and escape strategies in birds. *Behav. Ecol.* **12**, 150–156 (2001).
7. Meyers, J. J. & Herrel, A. Prey capture kinematics of ant-eating lizards. *J. Exp. Biol.* **208**, 113–127 (2005).
8. Bianco, I., Kampff, A. & Engert, F. Prey capture behavior evoked by simple visual stimuli in larval zebrafish. *Front. Syst. Neurosci.* **5**, 101 (2011).
9. Espinasa, L., Bibliowicz, J., Jeffery, W. R. & Rétaux, S. Enhanced prey capture skills in *Astyanax* cavefish larvae are independent from eye loss. *EvoDevo* **5**, 35 (2014).
10. Mischiati, M. et al. Internal models direct dragonfly interception steering. *Nature* **517**, 333–338 (2015).
11. Catania, K. C. Electric eels use high-voltage to track fast-moving prey. *Nat. Commun.* **6**, 8638 (2015).
12. Ben-Tov, M., Donchin, O., Ben-Shahar, O. & Segev, R. Pop-out in visual search of moving targets in the archer fish. *Nat. Commun.* **6**, 6476 (2015).
13. Borghuis, B. G. & Leonardo, A. The role of motion extrapolation in amphibian prey capture. *J. Neurosci.* **35**, 15430–15441 (2015).
14. Hoy, J. L., Yavorska, I., Wehr, M. & Niell, C. M. Vision drives accurate approach behavior during prey capture in laboratory mice. *Curr. Biol.* **26**, 3046–3052 (2016).
15. Villanueva, R., Perricone, V. & Fiorito, G. Cephalopods as predators: a short journey among behavioral flexibilities, adaptions, and feeding habits. *Front. Physiol.* **8**, 598 (2017).

16. Chang, C., Lim, Z. Y., Klomp, D. A., Norma-Rashid, Y. & Li, D. Aggressive spiders make the wrong decision in a difficult task. *Behav. Ecol.* **29**, 848–854 (2018).
17. Mearns, D. S., Donovan, J. C., Fernandes, A. M., Semmelhack, J. L. & Baier, H. Deconstructing hunting behavior reveals a tightly coupled stimulus-response loop. *Curr. Biol.* **30**, 54–69.e9 (2020).
18. Jiun-Shian Wu, J. & Chiao, C.-C. Switching by cuttlefish of preying tactics targeted at moving prey. *iScience* **26**, 108122 (2023).
19. Geberl, C., Brinkløv, S., Wiegrebe, L. & Surlykke, A. Fast sensory-motor reactions in echolocating bats to sudden changes during the final buzz and prey intercept. *Proc. Natl. Acad. Sci. USA* **112**, 4122–4127 (2015).
20. Wang, J. et al. Dynamic Decision Making in Predatory Pursuit of Mice. Available at SSRN: https://doi.org/10.2139/ssrn.4563035 (2023).
21. Pyke, G. H., Pulliam, H. R. & Charnov, E. L. Optimal foraging: a selective review of theory and tests. *Q. Rev. Biol.* **52**, 137–154 (1977).
22. Ngo, V. et al. Active vision during prey capture in wild marmoset monkeys. *Curr. Biol.* **32**, 3423–3428.e3 (2022).
23. Cisek, P. Resynthesizing behavior through phylogenetic refinement. *Atten. Percept. Psychophys.* **81**, 2265–2287 (2019).
24. Dukas, R. Evolutionary biology of animal cognition. *Annu. Rev. Ecol. Evol. Syst.* **35**, 347–374 (2004).
25. Mathis, A. et al. DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nat. Neurosci.* **21**, 1281–1289 (2018).
26. Wiltschko, A. B. et al. Mapping sub-second structure in mouse behavior. *Neuron* **88**, 1121–1135 (2015).
27. Robson, D. N. & Li, J. M. A dynamical systems view of neuroethology: uncovering stateful computation in natural behaviors. *Curr. Opin. Neurobiol.* **73**, 102517 (2022).
28. Miller, C. T. et al. Natural behavior is the language of the brain. *Curr. Biol.* **32**, R482–R493 (2022).
29. Ewert, J.-P. Neuroethology of releasing mechanisms: prey-catching in toads. *Behav. Brain Sci.* **10**, 337–368 (1987).
30. Szopa-Comley, A. W. & Ioannou, C. C. Responsive robotic prey reveal how predators adapt to predictability in escape tactics. *Proc. Natl Acad. Sci.* **119**, e2117858119 (2022).
31. Michaiel, A. M., Abe, E. T. & Niell, C. M. Dynamics of gaze control during prey capture in freely moving mice. *eLife* **9**, e57458 (2020).
32. Sunami, N. et al. Automated escape system: identifying prey's kinematic and behavioral features critical for predator evasion. *J. Exp. Biol*. **227**, jeb246772 (2024).
33. Horner, A. E. et al. The touchscreen operant platform for testing learning and memory in rats and mice. *Nat. Protoc.* **8**, 1961–1984 (2013).
34. Mueller-Paul, J. et al. Touchscreen performance and knowledge transfer in the red-footed tortoise (*Chelonoidis carbonaria*). *Behav. Process.* **106**, 187–192 (2014).
35. Kangas, B. D. & Bergman, J. Touchscreen technology in the study of cognition-related behavior. *Behav. Pharmacol.* **28**, 623–629 (2017).
36. Nieder, A. Evolution of cognitive and neural solutions enabling numerosity judgements: lessons from primates and corvids. *Philos. Trans. R. Soc. B Biol. Sci.* **373**, 20160514 (2018).
37. Oh, J., Šlipogor, V. & Fitch, W. T. Artificial visual stimuli for animal experiments: An experimental evaluation in a prey capture context with common marmosets (*Callithrix jacchus*). *J. Comp. Psychol.* **133**, 72–80 (2019).
38. Lewis, V., Laberge, F. & Heyland, A. Temporal profile of brain gene expression after prey catching conditioning in an anuran amphibian. *Front. Neurosci.* **13**, 1407 (2020).
39. Eisenberg, T. & Shein-Idelson, M. ReptiLearn: an automated home cage system for behavioral experiments in reptiles without human intervention. *PLoS Biol.* **22**, e3002411 (2024).
40. Schaerlaeken, V., Meyers, J. J. & Herrel, A. Modulation of prey capture kinematics and the role of lingual sensory feedback in the lizard *Pogona vitticeps*. *Zoology* **110**, 127–138 (2007).
41. Robins, A., Chen, P., Beazley, L. D. & Dunlop, S. A. Lateralized predatory responses in the ornate dragon lizard (*Ctenophorus ornatus*). *NeuroReport* **16**, 849–852 (2005).
42. Frohnwieser, A., Pike, T. W., Murray, J. C. & Wilkinson, A. Perception of artificial conspecifics by bearded dragons (*Pogona vitticeps*). *Integr. Zool.* **14**, 214–222 (2019).
43. Wasserman, E. A., Nagasaka, Y., Castro, L. & Brzykcy, S. J. Pigeons learn virtual patterned-string problems in a computerized touch screen environment. *Anim. Cogn.* **16**, 737–753 (2013).
44. Kleinberger, R., Cunha, J., McMahon, M. & Hirskyj-Douglas, I. No more angry birds: investigating touchscreen ergonomics to improve tablet-based enrichment for parrots. In *Proc. CHI Conference on Human Factors in Computing Systems* 1–16 (Association for Computing Machinery, 2024). https://doi.org/10.1145/3613904.3642119.
45. Harrison, R. A., Mohr, T. & van de Waal, E. Lab cognition going wild: Implementing a new portable touchscreen system in vervet monkeys. *J. Anim. Ecol.* **92**, 1545–1559 (2023).
46. Vale, G. L., Leinwand, J. G. & Joshi, P. B. Testing three primate species' attentional biases toward preferred and unpreferred foods: Seeing red or high valued food? *J. Comp. Psychol.* **138**, 177–189 (2024).
47. Yoo, S. B. M., Tu, J. C., Piantadosi, S. T. & Hayden, B. Y. The neural basis of predictive pursuit. *Nat. Neurosci.* **23**, 252–259 (2020).
48. Szabo, B., Noble, D. W. A. & Whiting, M. J. Learning in non-avian reptiles 40 years on: advances and promising new directions. *Biol. Rev.* **96**, 331–356 (2021).
49. Burghardt, G. Learning processes in reptiles. *Biol. Reptil.* **7**, 555–681 (1977).
50. Wilkinson, A. & Huber, L. Cold-blooded cognition: reptilian cognitive abilities. In *The Oxford Handbook of Comparative Evolutionary Psychology*. (Oxford Academics, 2012).
51. Burghardt, G. M. Environmental enrichment and cognitive complexity in reptiles and amphibians: Concepts, review, and implications for captive populations. *Appl. Anim. Behav. Sci.* **147**, 286–298 (2013).
52. De Meester, G. & Baeckens, S. Reinstating reptiles: from clueless creatures to esteemed models of cognitive biology. *Behaviour* **158**, 1057–1076 (2021).
53. Ashwood, Z. C. et al. Mice alternate between discrete strategies during perceptual decision-making. *Nat. Neurosci.* **25**, 201–212 (2022).
54. Qi, Y., Noble, D. W. A., Fu, J. & Whiting, M. J. Testing domain general learning in an Australian lizard. *Anim. Cogn.* **21**, 595–602 (2018).
55. Szabo, B., Noble, D. W. A., Byrne, R. W., Tait, D. S. & Whiting, M. J. Subproblem learning and reversal of a multidimensional visual cue in a lizard: evidence for behavioural flexibility? *Anim. Behav.* **144**, 17–26 (2018).
56. Szabo, B. & Whiting, M. J. A new protocol for investigating visual two-choice discrimination learning in lizards. *Anim. Cogn.* **25**, 935–950 (2022).
57. McNamee, D. & Wolpert, D. M. Internal models in biological control. *Annu. Rev. Control Robot. Auton. Syst.* **2**, 339–364 (2019).
58. Brenowitz, E. A. & Zakon, H. H. Emerging from the bottleneck: benefits of the comparative approach to modern neuroscience. *Trends Neurosci.* **38**, 273–278 (2015).
59. Yartsev, M. M. The emperor's new wardrobe: rebalancing diversity of animal models in neuroscience research. *Science* **358**, 466–469 (2017).
60. Laurent, G. On the value of model diversity in neuroscience. *Nat. Rev. Neurosci.* **21**, 395–396 (2020).
61. Regev, E., Albeck, N. & Shein-Idelson, M. PreyTouch: A touchscreen-based closed-loop system for studying predator-prey interactions. Zenodo https://doi.org/10.5281/zenodo.1419965.
62. Siegle, J. H. et al. Open Ephys: an open-source, plugin-based platform for multichannel electrophysiology. *J. Neural Eng.* **14**, 045003 (2017).
63. Peirce, J. et al. PsychoPy2: experiments in behavior made easy. *Behav. Res. Methods* **51**, 195–203 (2019).

Acknowledgements
This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant Agreement No. 949838) and from the Israel Science Foundation (ISF, grant No. 1133/20). The authors are most grateful to A. Shvartsman for technical and administrative assistance; the animal caretaker crew for lizard care; N. Leberstein, and M. Becker for comments on the manuscript; the Shein-Idelson laboratory for their suggestions during this work.

Author contributions
R.E. and M.S.I. initiated and designed the project. R.E. collected the behavioral data. N.A. collected the electrophysiological data. R.E. and M.S.I. analyzed the data, designed and performed the statistical analysis. R.E. and M.S.I. wrote the paper. R.E., N.A., and M.S.I. revised the manuscript. M.S.I. acquired the funding. M.S.I. supervised the work.

Competing interests
The authors declare no competing interests.

Additional information
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42003-024-07345-5.

Correspondence and requests for materials should be addressed to Mark Shein-Idelson.

Peer review information *Communications Biology* thanks Yuuki Kawabata, Ronen Segev, and Malachi Whitford for their contribution to the peer review of this work. Primary Handling Editor: Jasmine Pan.