nature machine intelligence

Article

https://doi.org/10.1038/s42256-024-00854-2

Discovering neural policies to drive behaviour by integrating deep reinforcement learning agents with biological neural networks

Received: 12 July 2023

Accepted: 10 May 2024

Published online: 14 June 2024

Check for updates

Chenguang Li [1] ✉, Gabriel Kreiman [2,3] ✉ & Sharad Ramanathan [4,5,6,7] ✉

Deep reinforcement learning (RL) has been successful in a variety of domains but has not yet been directly used to learn biological tasks by interacting with a living nervous system. As proof of principle, we show how to create such a hybrid system trained on a target-finding task. Using optogenetics, we interfaced the nervous system of the nematode *Caenorhabditis elegans* with a deep RL agent. Agents adapted to strikingly different sites of neural integration and learned site-specific activations to guide animals towards a target, including in cases where agents interfaced with sets of neurons with previously uncharacterized responses to optogenetic modulation. Agents were analysed by plotting their learned policies to understand how different sets of neurons were used to guide movement. Further, the animal and agent generalized to new environments using the same learned policies in food-search tasks, showing that the system achieved cooperative computation rather than the agent acting as a controller for a soft robot. Our system demonstrates that deep RL is a viable tool both for learning how neural circuits can produce goal-directed behaviour and for improving biologically relevant behaviour in a flexible way.

Guiding or improving animal behaviour directly through the nervous system is a common goal for neuroscience and robotics researchers alike[1–3]. Previous work in brain interfaces and animal robotics has attempted to use direct interventions to affect behaviour on a variety of tasks, relying on manual specification for stimulation frequencies, locations, dynamics and patterns[4–21]. A central difficulty with these approaches is that manual tuning has limited applicability, as it relies on knowledge of the neural circuits or mechanisms involved. Activation patterns for a given task and set of neurons are often unknown[7], nervous systems have complex intrinsic neural dynamics, and there is a combinatorial explosion of stimulation parameters to test. For direct neural stimulation, effective patterns can vary depending on which neurons are targeted and on the animal itself[22,23]. Thus, even though technologies for precise neuronal modulation exist[24,25], there lies the challenge of how to design an algorithm that can systematically and automatically learn strategies to activate a set of neurons to improve a particular behaviour[26–30].

Here we addressed this challenge using deep reinforcement learning (RL), assessing whether RL can autonomously integrate with an animal's nervous system to improve behaviour. In an RL setting, an

[1]Biophysics, Harvard University, Cambridge, MA, USA. [2]Boston Children's Hospital, Harvard Medical School, Boston, MA, USA. [3]Center for Brains, Minds and Machines, Cambridge, MA, USA. [4]Department of Molecular and Cellular Biology, Harvard University, Cambridge, MA, USA. [5]Center for Brain Science, Harvard University, Cambridge, MA, USA. [6]Department of Stem Cell and Regenerative Biology, Harvard University, Cambridge, MA, USA. [7]John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA. ✉e-mail: chenguang_li@fas.harvard.edu; gabriel.kreiman@childrens.harvard.edu; sharad@cgr.harvard.edu

agent collects rewards through interactions with its environment. By leveraging deep neural networks, RL algorithms have successfully discovered complex sequences of actions to solve a wide set of tasks[31–41]. These past successes relied on reward signals to train algorithms, a framework that can be adapted to biologically relevant goals, such as finding food or mates. Although other studies have incorporated machine learning into designing cyborg or biohybrid organisms[42–45], they have largely focused on optimizing only one means of interfacing with an animal, which could be difficult to scale up in neural interfaces, especially given the highly variable nature of living nervous systems. By using deep RL, we present instead a flexible framework that can, given only a reward signal, observations and a set of relevant actions, learn different ways of achieving a goal behaviour that adapt to the chosen interface.

We tested our ideas on the nematode *Caenorhabditis elegans*, interfacing an RL agent with its nervous system using optogenetic tools[24,27]. This animal has a small and accessible nervous system while still possessing a rich behavioural repertoire[46], making it a suitable candidate to test deep RL integration[47]. In a natural setting, *C. elegans* must navigate variable environments to avoid danger or find targets like food. Therefore, we aimed to build an RL agent that could learn how to interface with neurons to assist *C. elegans* in target-finding and food search. We tested the agent by connecting it to different sets of neurons with distinct roles in behaviour, where some of these neuronal sets did not have fully understood roles in directed movement. Agents could not only couple with different sets of neurons to perform a target-finding task but also generalize to improve food search across new environments in a zero-shot fashion: that is, without any prior training. We show that our neural–RL interface can be used to investigate the function of neural circuits in task performance, including with sets of neurons whose links to behaviours have not been previously established.

Connecting the nervous system to artificial intelligence and training agents

We used a closed-loop setup to couple an RL agent to an animal's nervous system (Fig. 1a,b). We formulated target-finding as an RL problem by defining a reward value as the negative distance of the animal's coordinates to a user-specified target (Fig. 1c; Methods). The RL agent's environment consisted of a ~1 mm adult animal and a 4-cm-diameter arena on an agar plate. Observations of the environment were given to the agent through a camera at 3 Hz, and features were automatically extracted from each camera frame to track the animal's centre of mass. During evaluation, target coordinates were subtracted from the animal's coordinates before being sent as part of the input to agents, (x_t, y_t). Head and body angles (θ_t^{body}, θ_t^{head}) were extracted from each frame relative to the $+x$ axis, and head angles were measured relative to body angles. We took polar coordinates of the angle measurements so that an observation was defined for every frame t, ($\sin\theta_t^{\text{body}}$, $\cos\theta_t^{\text{body}}$, $\sin\theta_t^{\text{head rel.}}$, $\cos\theta_t^{\text{head rel.}}$, x_t, y_t) (Fig. 1d). Each observation the agent received included these six variables from frames over the past 5 seconds, making agent inputs 90-dimensional at each timestep (6 variables × 3 frames per second (fps) × 5 seconds; Methods). These variables are relevant for the navigation task, although we note that other tasks may benefit from different sets of task-specific variables.

Given an observation at time t, the RL agent was trained to learn what action a_t to take at that time to maximize return, defined as a sum of rewards discounted over time (Fig. 1e and Methods). To take an action, the agent could decide whether to turn a light-emitting diode (LED) on or off at each timestep. Using optogenetics[24], the agent could modulate selected neurons that expressed either channelrhodopsin, a light-gated ion channel that can be stimulated by blue light (480 nm) to activate neurons[25], or archaerhodopsin, a light-sensitive proton pump that can be stimulated with green light (540 nm) to inhibit neurons.

We chose the soft actor–critic (SAC) algorithm for the RL agent because of its successes in simulated and real-world RL environments[37,41,48,49]. SAC has separate neural networks for a critic that learns to evaluate observations and an actor that learns to optimize actions based on critic evaluations for return maximization (Fig. 1f, Methods). Both networks take observations as input and consist of two layers with 64 units per layer (Methods). The actor network outputs probabilities of turning on the light at time t, $P(a_t = 1)$. We assigned the agent's action for that observation as 'light on' if the actor's output $P(a_t = 1) \geq 0.5$.

Deep RL tends to require large amounts of data. For instance, agents learning to play Atari can require thousands of hours of gameplay to achieve good performance[33,34]. It was infeasible to collect thousands of hours of recordings in our environment, and unlike video games or physical systems with reliable dynamics, adequate computer simulations of the *C. elegans* nervous system and its behaviours are not available to generate training data[50]. Therefore, to facilitate algorithm development and reduce the amount of data needed to learn the target-finding task, agents were trained offline on prerecorded data, collected for 20 min per animal for a total of 5 h. During training data collection, the light was turned on randomly with a probability of 0.1 every second (Fig. 1g, top, and Methods). Following approaches in supervised learning[51], the data were then augmented during training by randomly translating and rotating the animal in a virtual arena approximately the size of the 4-cm-diameter evaluation arena (Methods).

During training, deep RL agents were unstable and prone to sudden performance drops (Supplementary Fig. 1), similar to previous work[52,53]. In simulated environments, such performance crashes can be monitored using evaluation episodes in the exact environment used for testing. In our environment, evaluation episodes were impractical because they would have required many more times the amount of data than were used to train agents. Therefore, we tested several regularization methods to help with stability and found that ensembles of agents were effective for our environment (Supplementary Figs. 2–4). The final deep RL agents were ensembles of SAC agents, with the collection, training and evaluation pipeline shown in Fig. 1g. For lines 1–3 described in Supplementary Table 1, ensembles consisted of 20 agents. For lines 4–6, which exhibited less stable training dynamics, ensembles consisted of 30 agents (see Methods for training protocol). Supplementary Figs. 4 and 5 show examples of variation between independently trained agents and how ensembles stabilized agent policies.

Agents could navigate animals to targets

We first tested our system on the transgenic line *Pttx-3::ChR2*, referred to as line 1 in the text (Fig. 2a and Supplementary Table 1). In line 1, the *ttx-3* promoter drives expression of channelrhodopsin in AIY interneurons, which are known to be involved in chemotaxis. Prior work has established a deterministic strategy for navigating animals using optogenetic activation of AIY[26], used here as a 'human expert' standard to see whether our agent could achieve similar performance.

After training an RL agent on line 1, the agent was evaluated by placing an animal in the centre of a 4-cm-diameter arena and entering target coordinates as input to the agent (Fig. 2b). The agent was set to navigate the animal over a 10 min episode to a target placed in one of four possible locations. The agent learned a pattern of light activation (blue points) to manoeuvre the animal towards the target. A sample track of an animal driven by the agent to a target is in Fig. 2c (see also Supplementary Video 1). In contrast, when the light was off all the time (Fig. 2d) or turned on randomly (Fig. 2e and Supplementary Video 2), the animal fails to reach the target. For comparison, we considered the case where the light was turned on according to the known 'human expert' policy, which was also successful in driving the animal to the target (Fig. 2f). Figure 2g shows statistics for each condition: the closer the distance to the target, the better the performance. The agent's learned policy performed as well as the known policy, and both of those performed significantly better than controls

https://doi.org/10.1038/s42256-024-00854-2

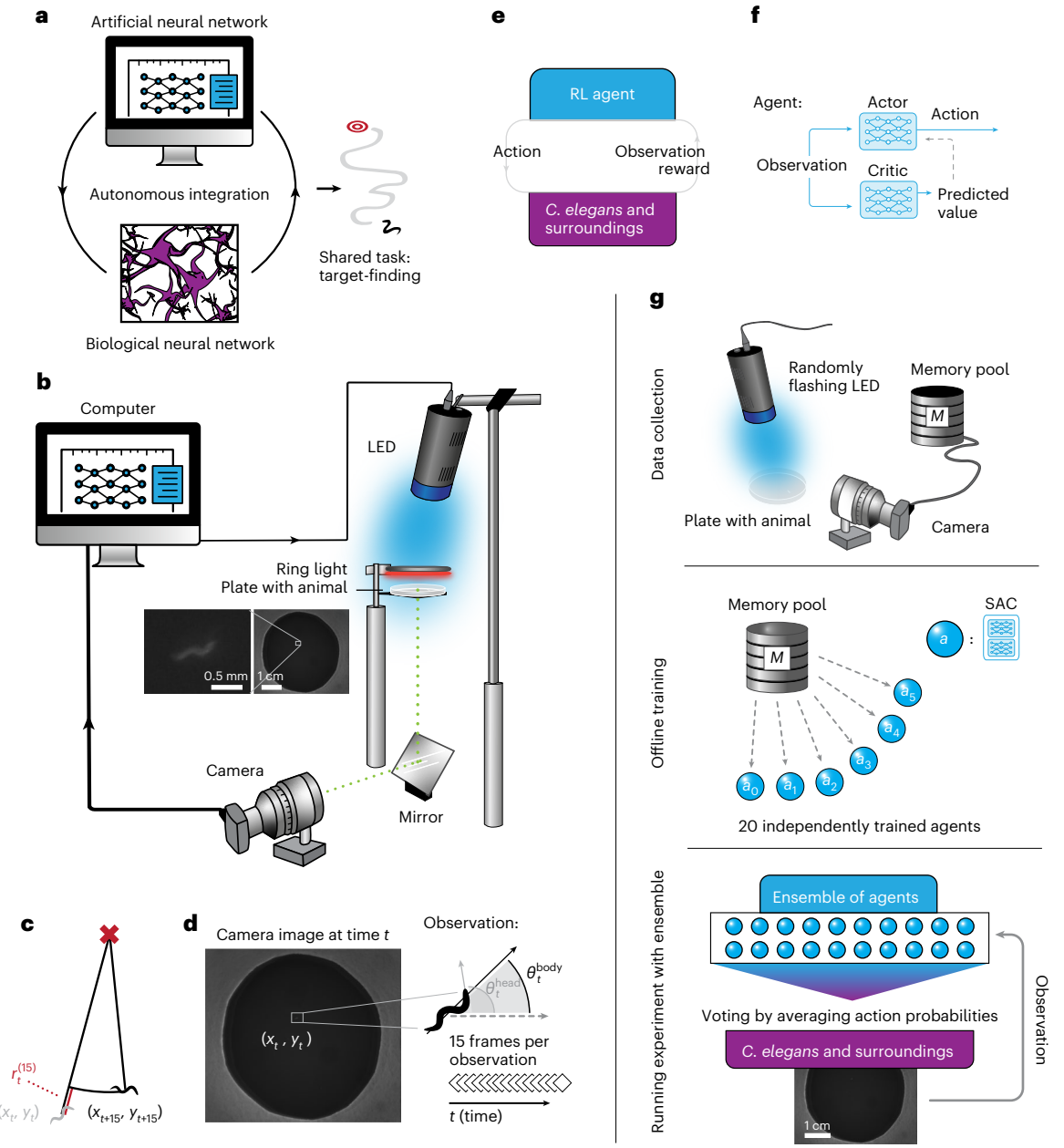

Fig. 1 | A system that integrates deep RL with the *C. elegans* neural network. **a**, Concept for combining artificial and biological neural networks for a shared task. **b**, Closed-loop setup using optogenetics. A single nematode was placed in a 4-cm-diameter field and illuminated by a red ring light for imaging. A camera and a high-powered LED (blue or green) were connected to a computer to form a closed-loop system. The LED modulated neurons carrying optogenetic constructs (see main text). **c**, Reward at time t, $r_t^{(15)}$, was defined as the change in distance to target between times t and $t+15$. **d**, Sample camera image at time t. An observation was a stack of six measurements from 15 frames (5 s at 3 fps) for a total of 90 variables per observation received by the agent at each timestep. Measurements were coordinates of the animal's centre of mass at time t (x_t, y_t) and the sines and cosines of the head and body angles, $(\theta_t^{\text{body}}, \theta_t^{\text{head}})$ of the animal relative to the positive x axis. **e**, RL loop diagram of the combined system. **f**, Actor–critic architecture used as a deep RL agent. **g**, Pipeline for training and evaluating the RL–animal system (see main text and Methods for details). A total of 5 h of data were collected where a light is flashed randomly on an animal stored in a memory pool (labelled *M*). Animals were switched out approximately every 20 min. Multiple soft actor–critic agents were independently trained on the memory pool. During evaluation, the agents were put into an ensemble that voted on actions in real time. Each individual agent's decision was based on the observation received from the camera.

(learned policy: $P = 0.00054$, no agent; $P = 0.00019$, random light. Known policy: $P = 0.0011$, no agent; $P = 0.00017$, random light). There was no significant difference in the time taken to reach within 0.5 cm of the target between the learned and known policies (Fig. 2g, inset; $P = 0.36$, one-sided Mann–Whitney U-test).

To understand what the agent trained on line 1 had learned[26], we sought a representative subspace of the 90-dimensional observation space in which to plot agent decisions. For every SAC agent in the ensemble, we plotted weights of the first layer of the actor network as a function of frame number to assess which input variables were associated with large weights (Fig. 2h and Supplementary Fig. 5). Head and body angles corresponding to the most recent frame in an observation (black arrows in Fig. 2h) had larger weight magnitudes than in earlier frames. Therefore, to visualize agent strategies, we fixed the 30 coordinate variables $((x_{t'}, y_{t'});\ t - 5\,\text{s} < t' < t)$ in each observation to a position left of the target (Fig. 2i and Methods) and plotted the

https://doi.org/10.1038/s42256-024-00854-2

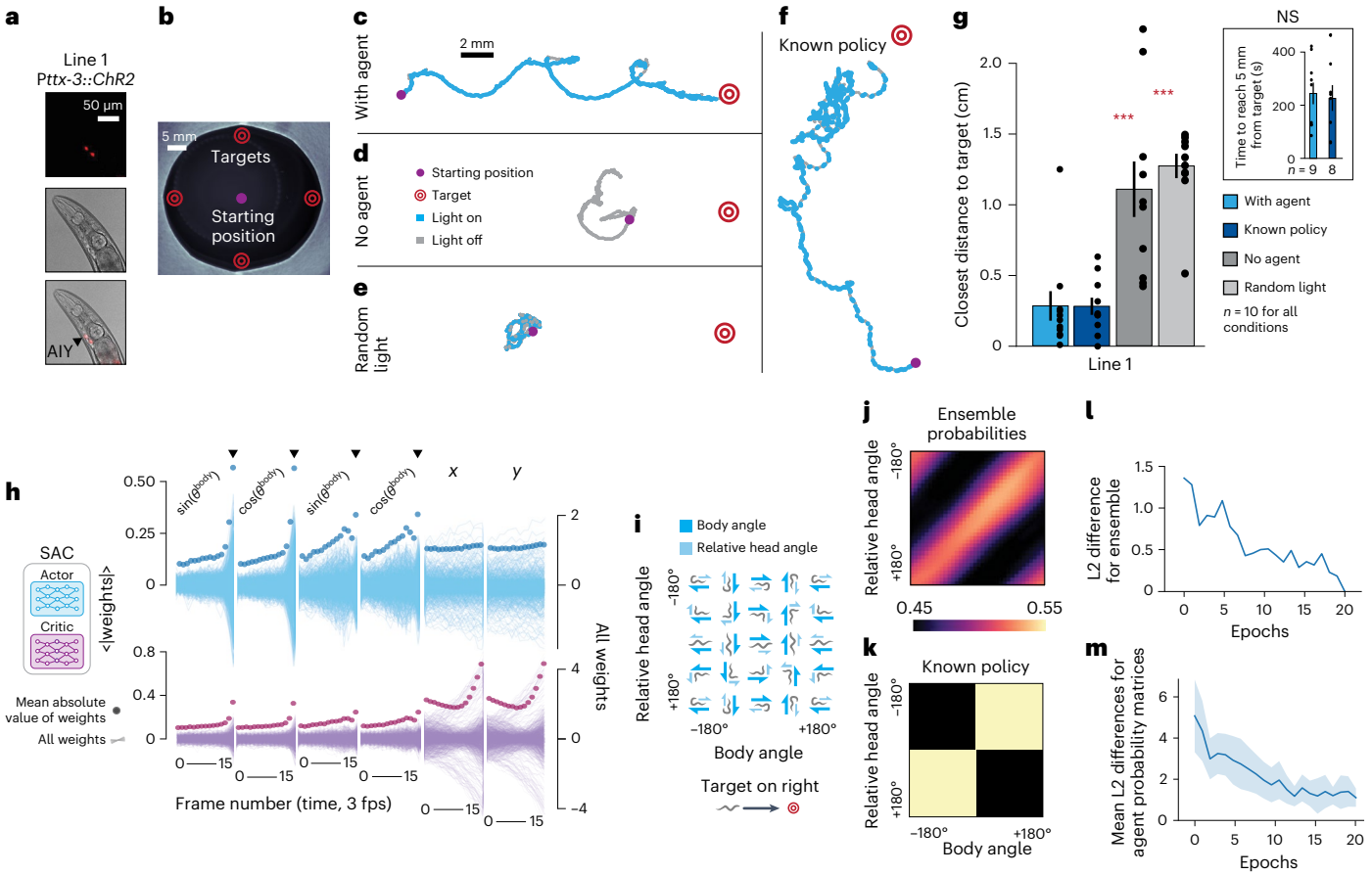

Fig. 2 | The system learned to navigate the *C. elegans* line 1 to a target. **a**, Optogenetically modified AIY neurons (black arrow) in line 1. **b**, Evaluation setup. The animal was placed in the centre (purple circle) of a filter paper circle with diameter 4 cm. In each 10 min episode, agents were tested on their ability to navigate the animal to one of the four target locations shown (red). **c**–**f**, Sample tracks with agent (**c**), without agent (**d**), with random light (**e**) and with a 'human expert' policy from literature (**f**)[26]. **g**, Closest distance to target achieved by animals for trials with and without an agent as well as with random light stimulations ($n = 10$ for each condition). Animals with agents moved significantly closer to targets than animals without agents. Plots show mean ± s.e.m. One-sided Mann–Whitney *U*-test, with agent versus with control conditions indicated by asterisks, $^{**}P < 0.01$, $^{***}P < 0.001$. (Learned policy: $P = 0.00054$, no agent; $P = 0.00019$, random light. Known policy: $P = 0.0011$, no agent; $P = 0.00017$, random light.) Times to reach within 0.5 cm of target for animals with learned and known policies were comparable, shown in inset (not significant (NS), $P = 0.36$, one-sided Mann–Whitney *U*-test). **h**, Weights of the first 64-neuron layer in all actor (top) and critic (bottom) networks in the agent ensemble. For angle-related variables, the most recent frames (black arrows) had the largest weights. **i**, Reference for the policy plots in **j** and **k**, showing example animal conformations. **j**, Trained agent probabilities for simulated inputs. **k**, The human expert policy plotted for comparison. It is similar to the learned agent policy but not identical. **l**, The L2 difference in the policy matrix between the final ensemble and ensembles at each epoch during training. By definition, the difference is 0 at epoch 20. **m**, Mean L2 differences between individual agents and the final ensemble, with standard deviation shaded in blue.

probability that the ensemble turned the light on as a function of body and head angles at the latest time in the observation (θ_t^{body}, θ_t^{head}) (Fig. 2j). The human expert policy is plotted in Fig. 2k using the same projection.

To interpret the policies, it is useful to compare Fig. 2i,j. The high-probability diagonal band in Fig. 2j corresponds to the same diagonal in Fig. 2i where the animal's head points towards the target. Interestingly, the agent's learned policy was conceptually similar but quantitatively different from the known expert policy in Fig. 2k, which placed greater emphasis on turning animals in the correct direction. Nonetheless, both policies were effective in the targeted navigation task.

The projection in Fig. 2j provided a way to plot agent training progress, with Fig. 2l,m showing the change in agent policies over 20 epochs of training. Figure 2l is the difference between the policy of full ensembles during and after training, and Fig. 2m takes differences between individual agent policies and compares them to the trained ensemble, plotting average differences with standard deviations. We saw that individual agents, even after training, could be quite far from the final policy, which highlighted the importance of ensembling.

Agents learned policies based on sites of integration

We aimed to build a robust and flexible algorithm that could adapt to its connected neurons. We next tested whether the RL agent could learn appropriate rules for a variety of neural connections without explicit prior knowledge about them. New agents were trained on five additional transgenic lines that were functionally distinct from line 1 and did not have associated human expert policies (Fig. 3). These lines are ordered in the text by agent performance compared to no light and random matched-frequency light controls. See Supplementary Table 1 and Fig. 3a for line genotypes and neuron expression.

Lines 3–6 expressed light-sensitive channels in multiple neuron types. Line 3 and 4 animals expressed archaerhodopsin, which inhibits neurons upon stimulation with green light (540 nm). Due to concerns about phototoxicity, agents for line 4 were restricted to short pulses

https://doi.org/10.1038/s42256-024-00854-2

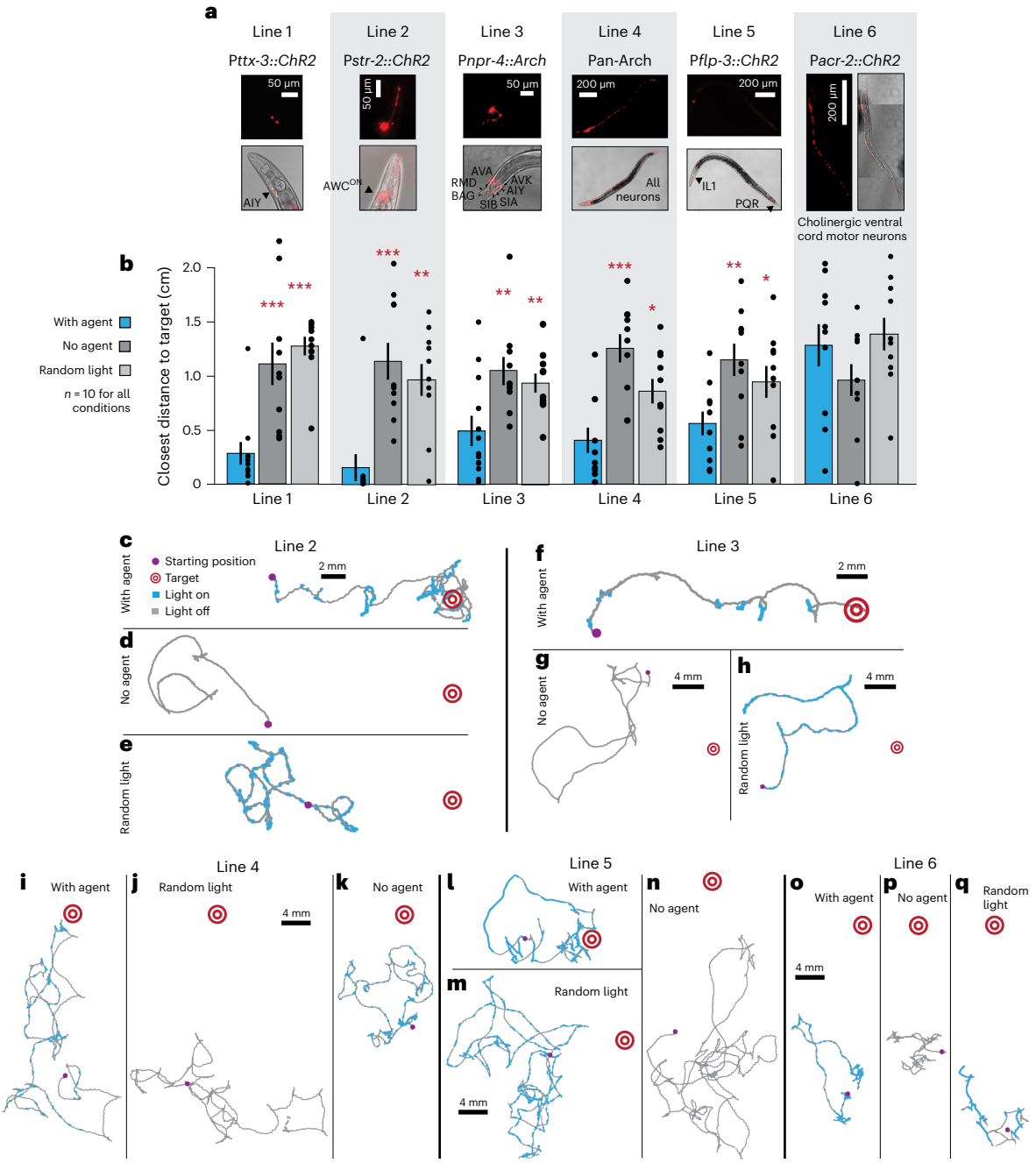

Fig. 3 | The system could successfully navigate different optogenetic lines to a target. a, Images of optogenetic lines with promoters and modified neurons. **b**, Statistics for each line ($n = 10$) comparing performance with agents, without agents and with frequency-matched random light controls, plotted as mean ± s.e.m. One-sided Mann–Whitney *U*-test, with agent versus with control conditions indicated by asterisks, $^{*}P < 0.05$, $^{**}P < 0.01$, $^{***}P < 0.001$. Lines 1–5 were successful. Line 1: $P = 0.00054$, no agent; $P = 0.00019$, random light. Line 2: $P = 0.0005$, no agent; $P = 0.0029$, random light. Line 3: $P = 0.0060$, no agent; $P = 0.0071$, random light. Line 4: $P = 0.0008$, no agent; $P = 0.0104$, random light. Line 5: $P = 0.0057$, no agent; $P = 0.03216$, random light. Line 6: $P = 0.9192$, no agent; $P = 0.4841$, random light. **c**–**e**, Following the format in Fig. 2c–f, example tracks for line 2 with positions of light activation along the trajectory highlighted in blue for animals with the agent (**c**), without any optogenetic activation (**d**) and with randomly flashing light (**e**). **f**–**q**, Example tracks for lines 3–6 for each experimental condition in **c**: line 3 with agent (**f**), without agent (**g**), random light (**h**); line 4 with agent (**i**), random light (**j**), no agent (**k**); line 5 with agent (**l**), random light (**m**), no agent (**n**); line 6 with agent (**o**), no agent (**p**), random light (**q**). Variability in starting positions for controls can be explained by free movement in the time between placing animals on the plate and starting the experiment, approximately 1 min.

during evaluation (Methods). These lines tested the abilities of the RL agent with different sets of neuronal connections and different means of modulation.

In lines 1–5, animals with trained agents moved closer to targets than control animals did (Fig. 3b). Example tracks showing agent activity during evaluation and controls are shown in Fig. 3c–q. Supplementary Videos 1–6 show agent performance and controls for lines 1–3, which performed best. Given that policies for goal-directed movement using optogenetic modulation of these lines were previously unknown, it was remarkable that agents still learned to direct these animals towards a target (for line 3, see ref. 54 for *npr-4* mutant behaviour; and for line 5, see ref. 55 for IL1 involvement in head withdrawal).

https://doi.org/10.1038/s42256-024-00854-2

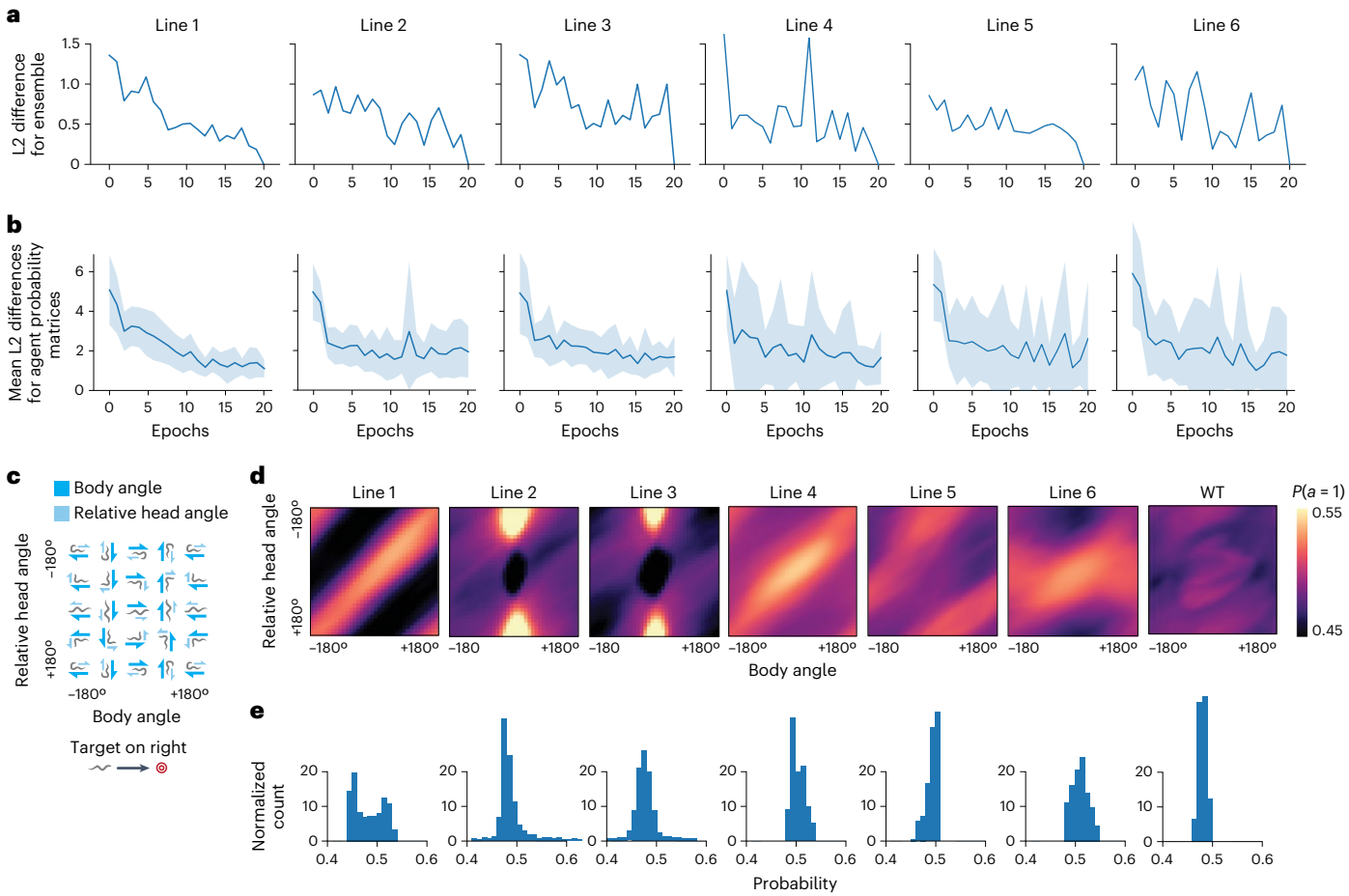

Fig. 4 | The system learned to navigate different optogenetic lines to a target with neuron-specific strategies. a, L2 differences between final ensembles and ensembles at each epoch during training. **b**, Mean L2 differences between individual agents in the ensemble and the final trained ensemble; shaded regions denote s.d. Within an ensemble, agents for lines 4–6 varied more than in lines 1–3, which is reflected in the narrower range of probabilities for Line 4–6 in **b**. **c**, The animal conformation reference plot for agent policies in **b** (repeated from Fig. 2i). **d**, All agent policies for Lines 1–6 and an agent trained on wild type data where there was no possible successful policy. Lines 1 and 4, as well as 2 and 3, had similar agent policies. **e**, Probabilities in **d** plotted as a histogram. Lines 1–3 had larger ranges, suggesting greater certainty. WT, wild type.

The agent successfully interacted with lines 3–5, which all involved multiple neurons (Fig. 3b), including line 4, which used the entire nervous system[56]. In this instance, the agents took advantage of increased movement after a period of freezing, in contrast to the line 3 policy that relied on slowing or turning during neuron inhibition. However, the agent failed to find an effective policy for line 6, where it was coupled to cholinergic muscle excitation in the ventral cord[57]. The standard deviation in the learned policy between agents in the ensembles was noticeably greater for lines 4–6 (Fig. 4a,b), which had poorer performance than lines 1–3 (Fig. 3b). Together these results show that the choice of sites of integration impact the performance of the animal–agent system.

We visualized policies using the metrics from Fig. 2i,j to understand how interfaced neurons were involved in target navigation. For reference, Fig. 4c shows animal postures used in mapping agent policies. Policies are plotted in Fig. 4d. Ensemble action certainty is also visible in Fig. 4d,e, in which lines 1–3 have probability values with a wider range than lines 4–6. This indicates agents are more certain about when to turn the light on or off in lines 1–3. For comparison, we show an agent trained on wild type animals (Fig. 4d) with no response to optogenetic modulation. The policies in Fig. 4d show that agents learned strategies tailored to the neurons they interfaced.

Agents predicted similarities between neural circuits

Broadly, there were three strategies represented by lines 1 and 4, lines 2 and 3 and line 5 (Fig. 4d). To understand how agent policies interacted with the nervous system, we focused on the most successful lines: 1, 2 and 3. Although the behaviour of line 1 in response to blue light is mostly to move forward and line 2 is mostly to reverse, policies were not merely inverses of each other. Rather, agents learned that line 1 control was dependent on the animal's head angle relative to the target, whereas Line 2 and 3 control depended on specific head and body angle combinations. Despite large differences in lines 2 and 3 (excitation of a single neuron in line 2 versus inhibition of multiple neurons in line 3), training on line 3 resulted in an action probability matrix that was remarkably similar to line 2.

To quantify these similarities in learned actions and to assess generalization across different sites of integration, we ran experiments where each agent was tested on each line (Fig. 5a). Sample tracks from combinations of agents and animals are shown in Fig. 5b with average results in Fig. 5c. To evaluate whether agent policies were predictive of cross-evaluation performance, we measured L2 norm differences of the action probability matrices (Fig. 5d). As intuitively observed in Fig. 5d, the policies from lines 2 and 3 are most similar. The corresponding plot using experimental data from Fig. 5c is shown in Fig. 5e. As expected,

https://doi.org/10.1038/s42256-024-00854-2

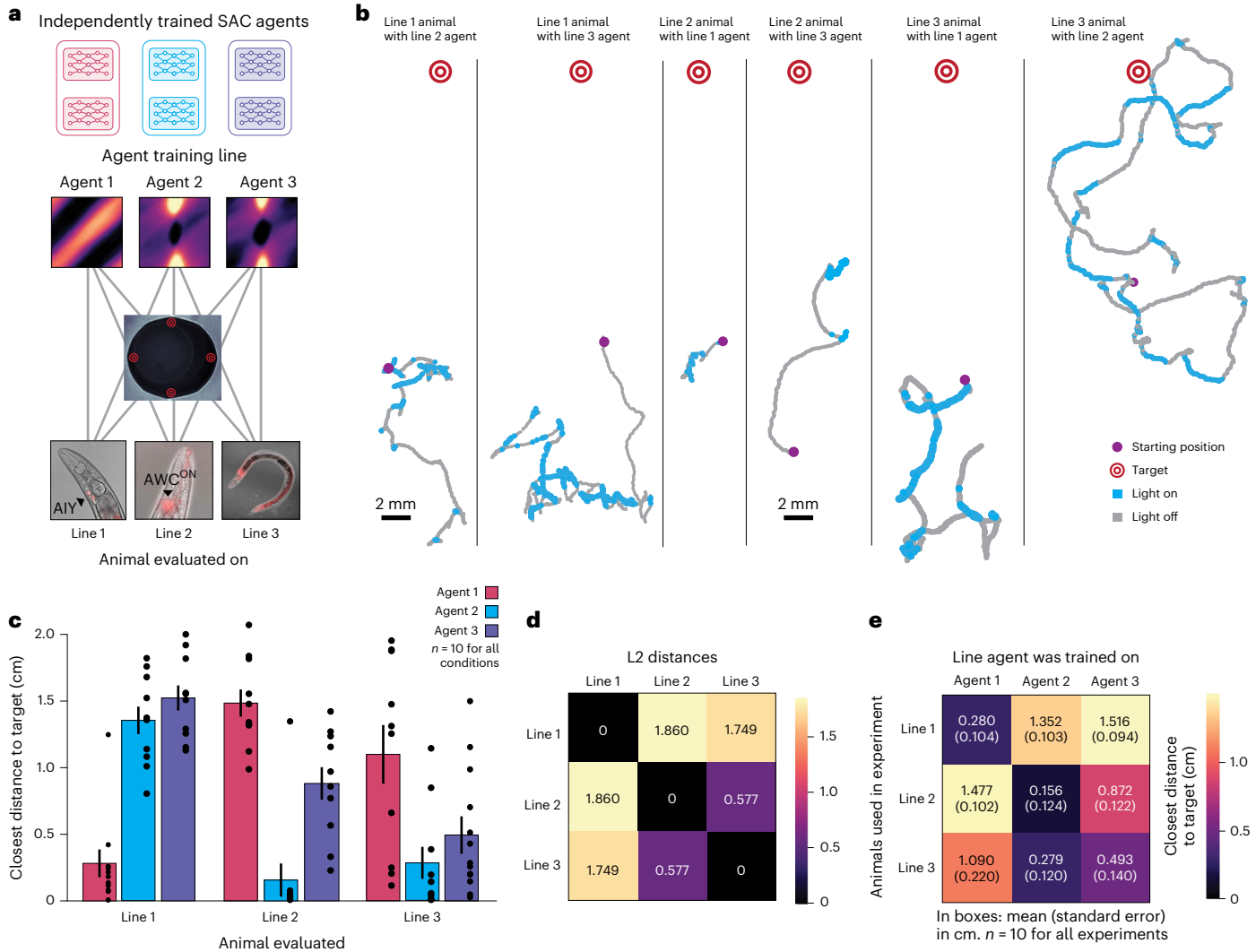

Fig. 5 | Agent policies can predict agent performance on other lines. a, An illustration of cross-evaluation experiments in which agents trained on each of the three best-performing lines were evaluated on every other line. **b**, Sample tracks with agent actions for each combination of agent and animal not shown in Fig. 2c or Fig. 3c,f. **c**, Statistics of closest distance to target for each combination of agent and animal, with $n = 10$ per condition. Data are presented as mean values ± s.e.m. **d**, L2 distances between ensemble action probability matrices for each genetic line. **e**, Mean closest distances (cm) to the target in a 10 min evaluation episode is shown with standard error in parentheses. Distances between the ensemble action probability matrices in **d** correlate with the closest distances achieved in across-policy evaluation experiments (Pearson's r, $r^2 = 0.8578$, $P = 0.000334$).

diagonal entries have low distances to targets; line 3 animals tested with line 2 agents also showed low distances.

Results in Fig. 5e correlated well with predictions based on the similarity of the action probability matrices in Fig. 5d ($r^2 = 0.8578$, $P = 0.000334$). As expected from the contrast in action probabilities in Fig. 4d, line 1 versus lines 2 and 3, line 1 did not respond well to agents trained on line 2 or 3. For example, when the agent trained on line 1 was tested with an animal from line 2, the closest distance reached from the target was about 1.477 ± 0.102 cm, much larger than when tested on line 1, 0.280 ± 0.104 cm (Fig. 5e). The closest distance was also comparable to or greater than the control conditions for line 2 (Fig. 3b), as the line 1 agent tended to drive line 2 animals away from rather than towards targets (P value < 0.08, no agent; P value < 0.009, random light; one-sided Mann–Whitney U-test). Likewise, neither line 2 nor 3 animals performed well on the task when paired with the line 1 agent. In summary, by comparing action probabilities learned by agents that were trained to couple to specific sets of neurons, we could make accurate predictions about the behaviour of these lines under optogenetic control in the target-finding task.

Another interesting finding was that line 2 and 3 animals were most successful when paired with the line 2 agent, even though the line 3 agent was trained on data from the line itself ($P < 0.002$, line 2 line with line 2 versus line 3 agent; $P < 0.04$, line 3 line with line 2 versus line 3 agent, one-sided Mann–Whitney U-test, $n = 10$). These results may be explained by higher data quality from the stronger response of line 2 to optogenetic stimulation (Supplementary Videos 1, 2, 5 and 6), reflected in greater action certainties in line 2 compared to line 3 (Fig. 4d). This suggests that training RL agents with less action noise could improve performance in noisy biological environments[58]. Overall, we demonstrate that our system can generate hypotheses about learning in biological environments, with greater access to internal mechanisms (through the artificial network) than an animal's nervous system alone can provide.

Animals corrected errors made by agents during food search

We aimed to see whether agents and animals could achieve tasks in a general way, integrating information flexibly just as animals can on

https://doi.org/10.1038/s42256-024-00854-2

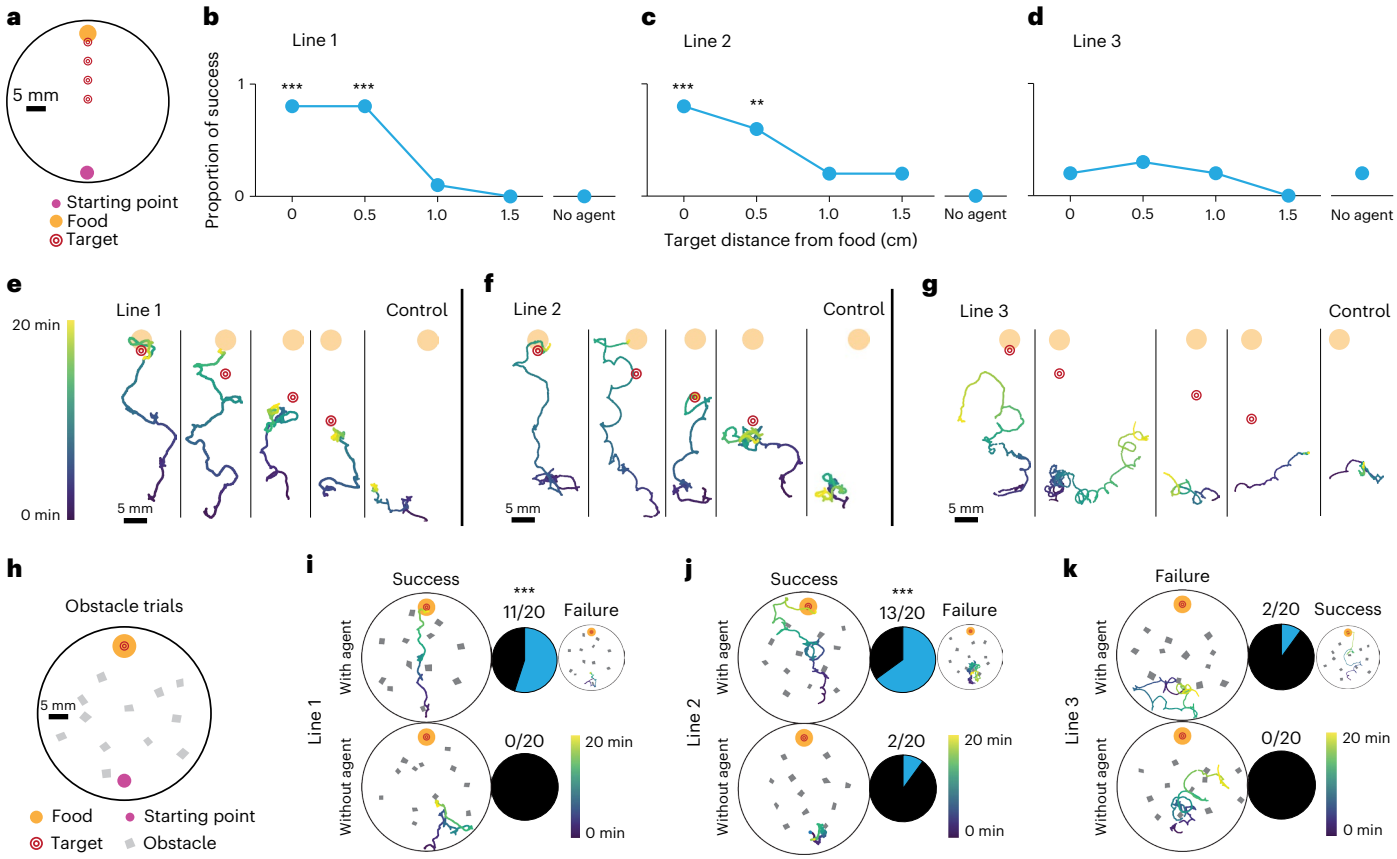

Fig. 6 | Animals with agents can correct errors and generalize to new situations. a, Error-handling food-search experiments. An animal was placed at the opposite end of a plate (large purple circle) from a 5 µm drop of OP50 *E. coli* bacteria (orange circle). Trials lasted 20 min and success was defined by whether animals reached food. Agents were directed to navigate animals a distance away from food (target location denoted by concentric red circles). **b**–**d**, Proportion of animals that reached food for line 1 (**b**), line 2 (**c**) and line 3 (**d**), $n = 10$ for each condition. For lines 1 and 2, targets up to 0.5 cm away led to significantly better performance than without agents. One-sided permutation tests; $^{**}P < 0.01$, $^{***}P < 0.001$ (with agent versus no agent; $P = 0.00034$ for line 1 with target at 0 and 0.5 cm from food and line 2 with target at 0 cm from food; $P = 0.0053$ for line 2 with target at 0.5 cm from food). **e**–**g**, Sample tracks for line 1 (**e**), line 2 (**f**) and line 3 (**g**) animals with agents based on the majority result of trials. Conditions without agents are shown in the fifth columns. **h**, Plate used for experiments with obstacles. Twelve paper quadrilaterals with side lengths approximately 2 mm were scattered on the plate. Agents were directed to navigate animals to food, and success was determined by whether animals reached food. **i**, Sample tracks for line 1 animals that succeeded (top left) or failed (top right), with control trials without agents (bottom). Success rates shown in pie charts. Animals with agents were significantly more likely to succeed; $^{***}P < 0.001$; one-sided permutation test; $P = 0.000076$. **j**, Sample tracks for line 2 animals: 13 of 20 animals succeeded with agents and 2 of 20 without. $^{***}P < 0.001$; one-sided permutation test; $P = 0.000407$. **k**, Sample tracks with line 3 animals. A failed trial in the top left represents the majority outcome: 2 of 20 animals reached food with agents and 0 of 20 without (one-sided permutation test, $P = 0.244$).

their own, so we next evaluated whether agents and animals could transfer abilities from the target-finding task to food search. Using the three best-performing lines, we tested whether animals could correct errors made by agents about the location of food. Targets for the artificial intelligence agent were placed at increasing distances from the edge of a 5 µl patch of food (OP50 *Escherichia coli* (*E. coli*) bacteria) to mimic errors made by the agent (Fig. 6a and Methods). Agents were on throughout the experiment, including after animals had reached the target. Animals were tested on whether they could reach food in 20 min trials with or without agents. Agents were identical to those used in Figs. 2–5, and each line was tested with its own agent. For lines 1 and 2, when targets were 0.5 cm away from food, animals could leave an agent's target region (a circle of radius 0.0625 cm; Methods) and moved to the food in eight of ten trials ($P < 0.0004$) (Fig. 6b,c). This was significantly different from trials without agent assistance, in which zero animals reached food in ten trials for both lines. Line 3 was not as successful with agent assistance (Fig. 6d), likely due to less reliable control (Fig. 3b). This suggests that simultaneous modulation of the neurons in this line is not as strongly linked to directed movement as in lines 1 and 2. In contrast, line 1 and 2 animals could switch between making decisions based on their own sensory systems or the agents, which were trained to keep animals at targets. Sample tracks for all experimental conditions are in Fig. 6e–g.

RL agents with animals could navigate new environments

We next tested whether the animal and agent could navigate an environment with obstacles to reach food, which represents a novel condition with a biologically relevant goal. We designed trials where 12 paper quadrilaterals with 1–3 mm edges (comparable to the 1 mm body length of *C. elegans*) were scattered randomly on the plate (Fig. 6h; Methods). Animals cannot cross these obstacles. We again tested animals on whether they could reach food during a 20 min trial, with and without agents. This was a challenging task because animals had to use their sensory and motor systems to navigate around obstacles, whereas agents had to navigate animals to food despite noisier movements.

Line 1 and 2 animals performed well in this new environment (Fig. 6i,j, P value < 0.0001, line 1; P value < 0.0004, line 2; permutation tests). Line 3 was not as successful (Fig. 6k); overall, agents could navigate line 3 animals closer to targets but could not achieve more difficult

food-search tasks. For lines 1 and 2, however, these data provide evidence that our system displays cooperative computation between artificial and biological neural networks to improve *C. elegans* food search in a zero-shot fashion in new environments.

Discussion

We presented a hybrid system that used deep RL to interact with an animal's nervous system to achieve a task following a reward signal. Agents customized themselves to specific and diverse sites of neural integration, and the combined system retained the animal's ability to flexibly integrate information in new environments. Importantly, we could use the same architecture and training process in all lines. Our results did not depend on the number of neurons that agents were interfaced with, nor whether interactions were excitatory or inhibitory, although a failure to learn (as in line 6) shows the importance of the particular neural circuit under control.

In previous work, brain–machine interfaces have allowed animals to control machines through neural recordings[2,59,60]. Conversely, supervised optogenetic manipulations have taken control of *C. elegans* neurons or muscles to turn the animal into a passive robot[26,61,62]. In contrast to both of these types of artificial–biological neural interactions, our work integrated a living nervous system with an artificial neural network, automatically discovered tailored neural activation patterns and did so in a way that allowed computations from both networks to drive animal behaviour.

We could then map out patterns of neural activity that were sufficient to drive specific behaviours, enabling us to learn about and compare the roles of different sets of neurons in producing the same behavioural outcome. Mapping out neural policies was possible not only for sets of neurons that were previously well-understood but also for sets that were not. We focused here on navigational tasks, which constitute a central aspect of worm behaviour, but our method for learning and visualizing agent policies can broadly be used to learn information about animal behaviour using other biologically relevant features besides the particular state space we chose.

It would be interesting for future work to test our method in larger state and action spaces, as one would find in animals with larger nervous systems than *C. elegans*. Deep RL has already solved complex simulated tasks in high-dimensional spaces with large numbers of parameters[31,33,35]. That, in addition to our work showing that deep RL is flexible to the site of integration, suggests its potential for use with larger animals whose nervous systems are more variable between individuals. Also, due to broad applicability of the RL framework, the algorithm can be applied to any other behavioural task with a measurable reward function. Overall, our study opens new avenues for using deep RL to understand neural circuits, train in biologically relevant real-world environments and modulate animal behaviour.

Methods

Animal genetics and care

Genetic lines. Strains are listed in Supplementary Table 1, available upon request. All animals had *lite-1* mutant backgrounds to reduce light sensitivity. Lines were chosen after an initial screen for response to optogenetic activation or inhibition.

Animal maintenance. *C. elegans* strains were cultured at 20 °C (room temperature) on nematode growth media (NGM) plates seeded with *E. coli* strain OP50. Animals used in optogenetic experiments were cultured at 20 °C on NGM plates seeded with *E. coli* strain OP50 with 1 mM all-trans-retinal (ATR) at a 9:1 volume ratio for at least 12 h before experiments. (ATR is a cofactor required for rhodopsin activity.)

Experimental setup

Experimental system hardware. Experiments were conducted at 20 °C. Two setups were built as in the diagram in Fig. 1b. The first used an Edmund Optics 5012 LE Monochrome USB 3.0 Lite Edition camera. The assay plate was lit with an Advanced Illumination RL1660 ring light. For the second rig, the camera was a USB-connected ThorLabs DCC1545M. Both cameras were run at 3 fps, which was a rate slow enough for image capture, image processing, action decision and action transmission to occur.

Lights for optogenetic illumination were Kessil PR160L LEDs at wavelengths of 467 nm for blue and 525 nm for green. The plate was illuminated with a Grandview COB Angel Eyes 110 mm Halo ring light. Kessil LEDs for optogenetic activation were controlled by a National Instruments DAQmx that was in turn managed through a Python library.

Animal tracking. For all experiments, animals were moved from food plates to a 10-cm-diameter NGM tracking plate. Tracking-plate setups depended on the experiment, but all plates had a filter paper ring to confine the animal to a 4-cm-diameter circle. We soaked the paper in 20 mM copper (II) chloride solution, an aversive substance to *C. elegans*, before placing it on the plates. Obstacles used in Fig. 6h–k were not soaked in copper solution. If food patches were used in the experiment, as in Fig. 6, 5 µl of OP50 *E. coli* bacteria were deposited on the plate and allowed to grow at room temperature (20 °C) for roughly 24 hours.

Collecting training data

Five hours of data were collected for each genetic line in 20 min episodes. In every episode, a single nematode cultured with ATR was placed on an NGM plate. As in the animal tracking setup, a filter paper barrier of diameter 4 cm was placed on the plate. A camera then recorded images at 3 fps while a blue or green LED flashed randomly on the plate. Blue light was used for animals modified with channelrhodopsin, and green light was used for animals modified with archaerhodopsin. A decision to turn the light on or off was made every 1 s with a probability of 10% on. If on, the light duration was also 1 s. Animals were switched out for new ones after each episode. Light decisions and images were stored for agent training in separate datasets for each line.

RL details

RL is a framework in which an agent interacts with an environment and attempts to maximize a reward signal. The agent receives observations from the environment, giving it an idea of the environment's current state, and learns what actions to take that will be most likely to maximize the reward signal received from the environment. The RL agent learns through experience an action probability distribution, $\pi(a_t|\boldsymbol{s}_t)$, where a_t is the action taken at time t, $\boldsymbol{s}_t$ is the state received from the environment corresponding to time t, and the maximized reward r_t is received at time t. Each of these variables is defined below.

We used a discrete SAC algorithm for all agents[41,48]. For each genetic line, 20 SAC agents were independently trained offline on the same data pool.

Variable definitions. *Observations*. Every camera image was preprocessed into features known to be relevant in *C. elegans* behaviour[26]. We used pixel coordinates (x,y) of the animal's centroid location in the image, with target coordinates subtracted from the centroid. During training, the target was always assumed to be (0,0), with coverage over the plate provided by random translations and rotations. Body angles were measured relative to the $+x$ axis and head angles relative to the body angle. Body angles were computed by fitting a line to a skeletonized worm image, and head angles were computed through template matching.

We performed head/tail identification by assigning the head label to the endpoint that was closest to the head endpoint in a previous frame. To handle reversals, a common behaviour in freely moving animals, the overall movement vector over 10 s was compared to tail-to-head vectors during the same window of time. If the vectors pointed in different directions, head and tail labels were switched.

Before each evaluation episode, 5 s of frames were collected to assign the first head label again by comparing movement vectors to tail-to-head vectors.

Angles were converted to sine and cosine pairs to avoid angle wraparound issues. Fifteen frames (5 s at 3 fps) were concatenated together for a single observation. Coordinates were normalized so their mean in each 15-frame observation was within [−0.5, 0.5]. An observation s_t corresponding to time t thus comprised 6 × 15 = 90 variables:

$$\mathbf{f}_t = \left(\sin\theta_t^{\text{body}}, \cos\theta_t^{\text{body}}, \sin\theta_t^{\text{head}}, \cos\theta_t^{\text{head}}, x_t, y_t\right)$$

$$\mathbf{s}_t = (\,\mathbf{f}_{t-14}, \mathbf{f}_{t-13}, \ldots, \mathbf{f}_t)$$

Above, f_t denotes the tuple of variables for the frame at time t. See Fig. 1d for a diagram defining the head and body angles.

Actions. An action at time t, a_t, was defined as a choice between the options 'light on' or 'light off', denoted by a binary 0 or 1 signal:

$$a_t \in \{0, 1\}$$

We did not place any constraints on actions, as all ensembles learned policies with overall light exposure that was under 50% of the time ('Standard evaluation').

Rewards. Reward r_t was based on the target-finding task and defined as the distance moved towards the target between the time of the action t and 15 frames (5 s) after the action (Fig. 1c):

$$r_t = \sqrt{(x_t - x_{\text{target}})^2 + (y_t - y_{\text{target}})^2} - \sqrt{(x_{t+15} - x_{\text{target}})^2 + (y_{t+15} - y_{\text{target}})^2}$$

A target region was defined as a circle of radius 30 pixels (625 µm). If the animal was within the target region, the calculated reward was replaced by a constant reward of 2. All other rewards were scaled by a factor of 2 to normalize values and facilitate training.

Training. As in standard RL, SAC searches for a policy $\pi(a_t|\boldsymbol{s}_t)$ for an environment with a transition distribution ρ_π. $\pi(a_t|\boldsymbol{s}_t)$ is the probability of taking an action a_t given an observation $\boldsymbol{s}_t$. Here we also make explicit the dependence of r_t on $\boldsymbol{s}_t$ and a_t. SAC deviates from the standard goal of maximizing the return or expected sum of rewards over time,

$$\sum_t \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[\gamma^t r_t(\mathbf{s}_t, a_t)\right].$$

Here, γ (fixed at 0.95) is a temporal discount factor that diminishes rewards far into the future. SAC maximizes not only the expected sum of rewards but also an entropy term weighted by a temperature parameter α:

$$\sum_t \mathbb{E}_{(\mathbf{s}_t,a_t)\sim\rho_\pi}\left[\gamma^t r_t(\mathbf{s}_t, a_t) + \alpha H(\pi(\bullet|\mathbf{s}_t))\right].$$

The sum now contains an added entropy term H of the policy $\pi(\bullet|\mathbf{s}_t)$, scaled by a temperature parameter α. $\pi(\bullet|\mathbf{s}_t)$ signifies the policy function π over all possible events. We used a discrete version of SAC with automatic entropy tuning (see code for implementation[63,64]).

Data augmentation. Once data were collected, they were stored in a memory buffer as tuples:

$$\mathbf{m}_t = (\mathbf{s}_t, a_t, r_t, \mathbf{s}_{t+15})$$

At each training step, a batch of 64 memory tuples were randomly drawn from the buffer and independently augmented by a random translation and rotation. First the tuple was centred such that the average of the location coordinates was at the origin, (0,0) pixels. Then a location within a ±900 pixel square (comparable to the size of the evaluation arena) was drawn from a uniform distribution and the coordinates recentred around that location. An angle was likewise chosen from a uniform distribution [0°,360°) and added to the measured angles in the memory tuple.

Training details. See Supplementary Table 2 for architecture and hyperparameter choices. Twenty agents per genetic line for lines 1–3 were trained independently on the same memory buffer for 20 epochs of 5,000 steps each. See Supplementary Fig. 5 for an example of training progression for individual agents on line 2. For lines 4–6, we found greater agent policy instability during training (Fig. 4b). In these cases, the animals' responses to optogenetic modulation were less tightly coupled to target navigation. We therefore trained 30 agents for lines 4–6. Each ensemble was trained for a minimum of 20 epochs of 5,000 steps. We then inspected policies visually to check that they satisfied two conditions. First, the ensemble policy needed to be non-trivial, or not always-on or always-off. Second, the policies needed to be fairly symmetric about the origin when plotted with body angles relative to target, as they should have been given the uniform random translations and rotations during training.

Minibatch size for all agents was 64. Weights were initialized using Xavier uniform initialization, and biases were initialized at 0. We tried dropout and weight decay on actors, critics or both and found that none of these regularizers helped enough to compensate for the need to choose more hyperparameters (see Supplementary Figs. 2–4).

Independent agents were trained such that the randomly taken action a_t, reward r_t and the associated states s_t and s_{t+15} were used to learn a state–action value function. This is called a Q-function and was learned by the critic network. The actor network then learned a policy that was the exponential of the Q-function. See ref. 41 for details.

Ensembles. Once the 20 agents for one ensemble were trained, they were combined by taking the average of their action probabilities and setting a threshold at 0.5. That is,

$$\pi_{\text{ensemble}}(a_t, |, \mathbf{s}_t) = \frac{1}{N}\sum_n^N \pi_n(a_t|\mathbf{s}_t)$$

where N = 20. If the average probability $\pi_{\text{ensemble}}(a_t, |, \mathbf{s}_t) \geq 0.5$, then the light was on at that timestep. Three to five random seeds were run for each genetic line, and the final ensemble was chosen based on inspection of visualized agent strategies.

Compute resources. All training was done on the FASRC Cannon cluster supported by the FAS Division of Science Research Computing Group at Harvard University. Every agent was trained on a compute node with one of the graphics processing units available on the cluster: Nvidia TitanX, K20m, K40m, K80, P100, A40, V100 or A100.

Agent strategy visualization. To visualize agent decisions, we simulated animal states in a smaller space than the full 90-dimensional inputs based on input weight magnitudes. Because the final timesteps of all angle measurements had larger magnitudes than previous timesteps (Fig. 2h and Supplementary Fig. 6), we chose to keep input angles constant within each observation and explored the full range of angle possibilities [−180°,180°) in increments of 10° for θ_t^{body} and θ_t^{head} (36 values each). The 30 coordinate variables $(x_{t'}, y_{t'})$; $t - 5 < t' < t$) were always fixed to 0.94 cm to the left of the target, which was exactly half the maximum distance used for random translations during training. In total, 36 head angle values × 36 body angle values gave rise to 1,296 different input observations, each of which was given to an agent ensemble that then output the decision probabilities recorded in the resultant action probability matrix.

Evaluation

All experiments involved a single animal placed on a 10-cm-diameter NGM plate with a 4-cm-diameter filter paper barrier soaked in copper (II) chloride. All animals were cultured on food with ATR and were thus sensitive to optogenetic perturbation. Animals were switched out for a new ones after each evaluation episode.

Standard evaluation. Animals were placed in the centre of the field. A target was randomly chosen among top, bottom, left and right options (Fig. 2b). The experiments with agents were run for 10 minutes each at 3 fps. At the end of the experiment, animals were switched out.

For controls without the agent, animals freely moved on the plate and were recorded for 10 min. A random target was assigned to compare controls to trials with agents.

For controls with random light exposure, the idea was to make sure that light exposure alone was not responsible for more movement, which could lead to an increased rate of success. Once all trials with agents had been run, the proportion of time where the light was on was calculated for each genetic line. These proportions were 0.4647 for line 1, 0.2896 for line 2 and 0.3844 for line 3. Animals were recorded while light decisions were made every 1 s, with the probability of light on according to the genetic lines listed.

For line 4 (Pan-Arch), due to concerns about phototoxicity, the evaluation was restricted to 1 s light pulses with 4 s rest periods between them.

Cross-agent evaluation. In Fig. 5, trained ensembles of agents were tested on the genetic lines they had not been trained on. The experiments were conducted identically to standard target-finding evaluations. Ten trials of 10 min each were performed for every agent–genetic line combination.

Error-handling food-search experiments. For the food-search experiments in Fig. 6a–g, a 10 cm NGM plate was prepared with a 4-cm-diameter filter paper circle soaked in 20 mM copper (II) chloride. Five µl of OP50 bacteria were grown for ~24 h before experiments.

Each trial lasted 20 min. An animal was placed on one end of the plate with the OP50 droplet at the opposite end. During the 20 min, the same agents trained on random data as in the standard evaluations were set to navigate animals to targets at 0, 0.5, 1 or 1.5 cm from the edge of the OP50 droplet. For control trials, agents were left off, and the animal roamed freely for 20 min.

Success was defined as a binary outcome as in the obstacle experiments. If an animal reached the food within the 20 min trial, it was counted as a success. Out of 270 trials run across all genetic lines involving OP50 droplets (obstacles and food search), only one Line 1 animal left food after reaching it during a food-search trial when the target was placed 1 cm from the food edge. This trial was counted as a success.

Obstacle food-search experiments. For the obstacle trials in Fig. 6h–k, a 10 cm NGM plate was prepared with a 4-cm-diameter filter paper ring soaked in a 20 mM copper (II) chloride solution. We cut 12 pieces of filter paper into quadrilaterals with side lengths 1–3 mm and scattered them on the plate (they were not soaked in copper (II) chloride solution). Sample arrangements are shown in Fig. 6h–k. Plates were replaced with new obstacle arrangements every 5–10 trials. Five µl of OP50 bacteria were grown on one side of the plate for ~24 h before experiments.

Each obstacle experiment was a 20 min trial. A single animal was placed on one end of the plate as in Fig. 6h, with the food droplet on the other end and the obstacles between animal and food. Trained agents (the same agent ensembles used in standard evaluations) were run on the genetic line they were trained on for 20 min. Agents were not retrained to handle obstacles. Control trials had no optogenetic manipulation; that is, the animal was allowed to freely roam the plate with obstacles and food for 20 min. Success was defined as a binary outcome, indicating whether an animal reached food during the trial.

Reporting summary

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

Data availability

All processed animal tracks used to generate figures are available at https://github.com/ccli3896/RLWorms.git (ref. 63).

Code availability

Analysis code and training code are available at https://github.com/ccli3896/RLWorms.git (ref. 63) and https://doi.org/10.5281/zenodo.11002033 (ref. 65).

References

1. Romano, D., Donati, E., Benelli, G. & Stefanini, C. A review on animal–robot interaction: from bio-hybrid organisms to mixed societies. *Biol. Cybern.* **113**, 201–225 (2019).
2. Tankus, A., Fried, I. & Shoham, S. Cognitive-motor brain–machine interfaces. *J. Physiol. Paris* **108**, 38–44 (2014).
3. Bostrom, N. & Sandberg, A. Cognitive enhancement: methods, ethics, regulatory challenges. *Sci. Eng. Ethics* **15**, 311–341 (2009).
4. Afraz, S.-R., Kiani, R. & Esteky, H. Microstimulation of inferotemporal cortex influences face categorization. *Nature* **442**, 692–695 (2006).
5. Bonizzato, M. & Martinez, M. An intracortical neuroprosthesis immediately alleviates walking deficits and improves recovery of leg control after spinal cord injury. *Sci. Transl. Med.* **13**, eabb4422 (2021).
6. Enriquez-Geppert, S., Huster, R. J. & Herrmann, C. S. Boosting brain functions: Improving executive functions with behavioral training, neurostimulation, and neurofeedback. *Int. J. Psychophysiol.* **88**, 1–16 (2013).
7. Iturrate, I., Pereira, M., Millán, J. & del, R. Closed-loop electrical neurostimulation: challenges and opportunities. *Curr. Opin. Biomed. Eng.* **8**, 28–37 (2018).
8. Lafer-Sousa, R. et al. Behavioral detectability of optogenetic stimulation of inferior temporal cortex varies with the size of concurrently viewed objects. *Curr. Res. Neurobiol.* **4**, 100063 (2023).
9. Lu, Y. et al. Optogenetically induced spatiotemporal gamma oscillations and neuronal spiking activity in primate motor cortex. *J. Neurophysiol.* **113**, 3574–3587 (2015).
10. Salzman, D. C., Britten, K. H. & Newsome, W. T. Cortical microstimulation influences perceptual judgements of motion direction. *Nature* **346**, 174–177 (1990).
11. Schild, L. C. & Glauser, D. A. Dual color neural activation and behavior control with Chrimson and CoChR in Caenorhabditis elegans. *Genetics* **200**, 1029–1034 (2015).
12. Xu, J. et al. Thalamic stimulation improves postictal cortical arousal and behavior. *J. Neurosci.* **40**, 7343–7354 (2020).
13. Park, S.-G. et al. Medial preoptic circuit induces hunting-like actions to target objects and prey. *Nat. Neurosci.* **21**, 364–372 (2018).
14. Yang, J., Huai, R., Wang, H., Lv, C. & Su, X. A robo-pigeon based on an innovative multi-mode telestimulation system. *Biomed. Mater. Eng.* **26**, S357–S363 (2015).
15. Holzer, R. & Shimoyama, I. Locomotion control of a bio-robotic system via electric stimulation. In *Proc. Institute of Electrical and Electronics Engineers/Robotics Society of Japan International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications* 1514–1519 (IEEE, 1997).

16. Talwar, S. K. et al. Rat navigation guided by remote control. *Nature* **417**, 37–38 (2002).
17. Sato, H. et al. A cyborg beetle: insect flight control through an implantable, tetherless microsystem. In *Proc. 21st Institute of Electrical and Electronics Engineers International Conference on Micro Electro Mechanical Systems* 164–167 (IEEE, 2008); https://doi.org/10.1109/MEMSYS.2008.4443618
18. Peckham, P. H. & Knutson, J. S. Functional electrical stimulation for neuromuscular applications. *Annu. Rev. Biomed. Eng.* **7**, 327–360 (2005).
19. Kashin, S. M., Feldman, A. G. & Orlovsky, G. N. Locomotion of fish evoked by electrical stimulation of the brain. *Brain Res.* **82**, 41–47 (1974).
20. Hinterwirth, A. J. et al. Wireless stimulation of antennal muscles in freely flying Hawkmoths leads to flight path changes. *PLoS ONE* **7**, e52725 (2012).
21. Sanchez, C. J. et al. Locomotion control of hybrid cockroach robots. *J. R. Soc. Interface* **12**, 20141363 (2015).
22. Bergmann, E., Gofman, X., Kavushansky, A. & Kahn, I. Individual variability in functional connectivity architecture of the mouse brain. *Commun. Biol.* **3**, 1–10 (2020).
23. Mueller, S. et al. Individual variability in functional connectivity architecture of the human brain. *Neuron* **77**, 586–595 (2013).
24. Husson, S. J., Gottschalk, A. & Leifer, A. M. Optogenetic manipulation of neural activity in C. elegans: from synapse to circuits and behaviour. *Biol. Cell* **105**, 235–250 (2013).
25. Nagel, G. et al. Channelrhodopsin-2, a directly light-gated cation-selective membrane channel. *Proc. Natl Acad. Sci. USA* **100**, 13940–13945 (2003).
26. Kocabas, A., Shen, C.-H., Guo, Z. V. & Ramanathan, S. Controlling interneuron activity in Caenorhabditis elegans to evoke chemotactic behaviour. *Nature* **490**, 273–277 (2012).
27. Leifer, A. M., Fang-Yen, C., Gershow, M., Alkema, M. J. & Samuel, A. D. T. Optogenetic manipulation of neural activity in freely moving Caenorhabditis elegans. *Nat. Methods* **8**, 147–152 (2011).
28. Wen, Q. et al. Proprioceptive coupling within motor neurons drives C. elegans forward locomotion. *Neuron* **76**, 750–761 (2012).
29. Hernandez-Nunez, L. et al. Reverse-correlation analysis of navigation dynamics in Drosophila larva using optogenetics. *eLife* **4**, e06225 (2015).
30. Donnelly, J. L. et al. Monoaminergic orchestration of motor programs in a complex C. elegans behavior. *PLoS Biol.* **11**, e1001529 (2013).
31. Silver, D. et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484–489 (2016).
32. Silver, D. et al. Mastering the game of Go without human knowledge. *Nature* **550**, 354–359 (2017).
33. Schrittwieser, J. et al. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature* **588**, 604–609 (2020).
34. Mnih, V. et al. Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).
35. Vinyals, O. et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* **575**, 350–354 (2019).
36. Berner, C. et al. Dota 2 with large scale deep reinforcement learning. Preprint at http://arxiv.org/abs/1912.06680 (2019).
37. Wurman, P. R. et al. Outracing champion Gran Turismo drivers with deep reinforcement learning. *Nature* **602**, 223–228 (2022).
38. Degrave, J. et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature* **602**, 414–419 (2022).
39. Ibarz, J. et al. How to train your robot with deep reinforcement learning: lessons we have learned. *Int. J. Rob. Res.* **40**, 698–721 (2021).
40. Haydari, A. & Yılmaz, Y. Deep reinforcement learning for intelligent transportation systems: a survey. *IEEE Trans. Intell. Transp. Syst.* **23**, 11–32 (2022).
41. Haarnoja, T., Zhou, A., Abbeel, P. & Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proc. 35th International Conference on Machine Learning* 1861–1870 (PMLR, 2018).
42. Yang, X., Jiang, X.-L., Su, Z.-L. & Wang, B. Cyborg moth flight control based on fuzzy deep learning. *Micromachines* **13**, 611 (2022).
43. Ariyanto, M., Refat, C. M. M., Hirao, K. & Morishima, K. Movement optimization for a cyborg cockroach in a bounded space incorporating machine learning. *Cyborg Bionic Syst.* **4**, 0012 (2023).
44. Zheng, N. et al. Real-time and precise insect flight control system based on virtual reality. *Electron. Lett.* **53**, 387–389 (2017).
45. Zheng, N. et al. Abdominal-waving control of tethered bumblebees based on sarsa with transformed reward. *IEEE Trans. Cybern.* **49**, 3064–3073 (2019).
46. Ardiel, E. L. & Rankin, C. H. An elegant mind: learning and memory in Caenorhabditis elegans. *Learn. Mem.* **17**, 191–201 (2010).
47. Kim, J. & Shlizerman, E. Deep reinforcement learning for neural control. Preprint at https://arxiv.org/abs/2006.07352 (2020).
48. Christodoulou, P. Soft actor-critic for discrete action settings. Preprint at https://arxiv.org/abs/1910.07207 (2019).
49. Wong, C.-C., Chien, S.-Y., Feng, H.-M. & Aoyama, H. Motion planning for dual-arm robot based on soft actor-critic. *IEEE Access* **9**, 26871–26885 (2021).
50. Sarma, G. P. et al. OpenWorm: overview and recent advances in integrative biological simulation of Caenorhabditis elegans. *Phil. Trans. R. Soc. B* **373**, 20170382 (2018).
51. Shorten, C. & Khoshgoftaar, T. M. A survey on image data augmentation for deep learning. *J. Big Data* **6**, 60 (2019).
52. Nikishin, E. et al. Improving stability in deep reinforcement learning with weight averaging. *Uncertainty in Artificial Intelligence Workshop on Uncertainty in Deep Learning* (2018).
53. Stable Baselines 2.10.2 documentation. *Reinforcement Learning Resources* https://stable-baselines.readthedocs.io/en/master/guide/rl.html (2021).
54. Bhardwaj, A., Thapliyal, S., Dahiya, Y. & Babu, K. FLP-18 functions through the G-protein-coupled receptors NPR-1 and NPR-4 to modulate reversal length in Caenorhabditis elegans. *J. Neurosci.* **38**, 4641–4654 (2018).
55. Riddle, D. L., Blumenthal, T., Meyer, B. J. & Priess, J. R. *Mechanosensory Control of Locomotion*. *C. elegans II* 2nd edn (Cold Spring Harbor Laboratory Press, 1997).
56. Brandt, R., Gergou, A., Wacker, I., Fath, T. & Hutter, H. A Caenorhabditis elegans model of tau hyperphosphorylation: induction of developmental defects by transgenic overexpression of Alzheimer's disease-like modified tau. *Neurobiol. Aging* **30**, 22–33 (2009).
57. Jospin, M. et al. A neuronal acetylcholine receptor regulates the balance of muscle excitation and inhibition in Caenorhabditis elegans. *PLoS Biol.* **7**, e1000265 (2009).
58. Hollenstein, J., Auddy, S., Saveriano, M., Renaudo, E. & Piater, J. Action noise in off-policy deep reinforcement learning: Impact on exploration and performance. *Transactions on Machine Learning Research* (2022); https://openreview.net/forum?id=NljBlZ6hmG
59. Andersen, R. A., Aflalo, T., Bashford, L., Bjånes, D. & Kellis, S. Exploring cognition with brain–machine interfaces. *Annu. Rev. Psychol.* **73**, 131–158 (2022).
60. Sussillo, D., Stavisky, S. D., Kao, J. C., Ryu, S. I. & Shenoy, K. V. Making brain–machine interfaces robust to future neural variability. *Nat. Commun.* **7**, 1–13 (2016).
61. Dong, X. et al. Toward a living soft microrobot through optogenetic locomotion control of Caenorhabditis elegans. *Sci. Robot.* **6**, eabe3950 (2021).

62. Tandon, P. pytorch-soft-actor-critic. *GitHub* https://github.com/pranz24/pytorch-soft-actor-critic (2022).
63. Li, C. RLWorms. *GitHub* https://github.com/ccli3896/RLWorms.git (2024).
64. Kazemipour, A. Discrete SAC PyTorch, *GitHub*, https://github.com/alirezakazemipour/Discrete-SAC-PyTorch (2020).
65. Li, C. RLWorms. *Zenodo* https://doi.org/10.5281/zenodo.11002033 (2024).

Acknowledgements

We thank S. Bhupatiraju for discussions about RL and comments on the manuscript. We thank T. Hallacy and A. Yonar for guidance in *C. elegans* experiments and C. McCartan for input on statistical analyses. We would like to thank Dr. Jeffrey Lee for providing us with customized high power LED light sources. We thank K. Blum, C. Pehlevan, G. Anand, A. Bacanu, B. Brissette, D. Hidalgo, R. Huang, H. Megale, W. Weiter, Y. Ilker Yaman, V. Zhuang and S. Zwick for comments on the manuscript. This work was supported in part by National Institute of General Medical Sciences grant no. 1R01NS117908-01 (S.R.), the Dean's Competitive Fund from Harvard University (S.R., C.L.), National Institutes of Health grant no. R01EY026025 (G.K.), the Fetzer Foundation (G.K.) and a National Science Foundation Graduate Research Fellowship Program fellowship (C.L.).

Author contributions

All the authors designed the study. C.L. wrote code, performed experiments and did data analysis. All the authors wrote the manuscript.

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/s42256-024-00854-2.

Correspondence and requests for materials should be addressed to Chenguang Li, Gabriel Kreiman or Sharad Ramanathan.

Peer review information *Nature Machine Intelligence* thanks Artur Luczak and Greg Wayne for their contribution to the peer review of this work.

nature portfolio

Corresponding author(s): Chenguang Li, Gabriel Kreiman, Sharad Ramanathan

Last updated by author(s): Aug 15, 2023

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐	A description of all covariates tested
☒	☐	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☐	☒	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	Custom code available at public repository stated in manuscript. https://github.com/ccli3896/RLWorms.git
Data analysis	Custom code available at public repository stated in manuscript. https://github.com/ccli3896/RLWorms.git

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

Data publicly available at repository stated in manuscript. https://github.com/ccli3896/RLWorms.git

Human research participants

Policy information about studies involving human research participants and Sex and Gender in Research.

Reporting on sex and gender	*Use the terms sex (biological attribute) and gender (shaped by social and cultural circumstances) carefully in order to avoid confusing both terms. Indicate if findings apply to only one sex or gender; describe whether sex and gender were considered in study design whether sex and/or gender was determined based on self-reporting or assigned and methods used. Provide in the source data disaggregated sex and gender data where this information has been collected, and consent has been obtained for sharing of individual-level data; provide overall numbers in this Reporting Summary. Please state if this information has not been collected. Report sex- and gender-based analyses where performed, justify reasons for lack of sex- and gender-based analysis.*
Population characteristics	*Describe the covariate-relevant population characteristics of the human research participants (e.g. age, genotypic information, past and current diagnosis and treatment categories). If you filled out the behavioural & social sciences study design questions and have nothing to add here, write "See above."*
Recruitment	*Describe how participants were recruited. Outline any potential self-selection bias or other biases that may be present and how these are likely to impact results.*
Ethics oversight	*Identify the organization(s) that approved the study protocol.*

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	Sample sizes were chosen by heuristic to demonstrate reliable effects.
Data exclusions	Animals were excluded as mentioned in Methods section if they were immobile throughout an evaluation or collection episode.
Replication	Results were replicable between trials.
Randomization	Allocation into control/evaluation groups was random.
Blinding	Blinding was not done to keep track of animal lines. Metrics were collected and analyzed immediately using a common pipeline and scripts between animal lines and control/experiment trials.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☐	☒ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging

Animals and other research organisms

Policy information about studies involving animals; ARRIVE guidelines recommended for reporting animal research, and Sex and Gender in Research

Laboratory animals	Caenorhabditis elegans strains: sraEx281, sraEx301, sraEx352, sraEx446, sraEx336, sraEx437, adult hermaphrodites. All available upon request.
Wild animals	No wild animals.
Reporting on sex	All animals were hermaphrodites.
Field-collected samples	No samples from field.
Ethics oversight	No ethical approval required for C. elegans research.

Note that full information on the approval of the study protocol must also be provided in the manuscript.