Integrating artificial and biological neural networks to improve animal task performance using deep reinforcement learning

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

[1] Biophysics Program, Harvard University, Cambridge, MA 02138, USA
[2] Department of Molecular and Cellular Biology, Harvard University, Cambridge, MA 02138, USA
[3] Boston Children's Hospital, Harvard Medical School, Boston, MA 02115
[4] Center for Brains, Minds and Machines, Cambridge, MA 02142
[5]Center for Brain Science, Harvard University, Cambridge MA 02138
[6] Department of Stem Cell and Regenerative Biology, Harvard University, Cambridge, MA 02138, USA
[7] John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, USA
[†] To whom correspondence should be addressed. Email: chenguang_li@fas.harvard.edu, gabriel.kreiman@childrens.harvard.edu, sharad@cgr.harvard.edu

Abstract

Artificial neural networks have performed remarkable feats in a wide variety of domains. However, artificial intelligence algorithms lack the flexibility, robustness, and generalization power of biological neural networks. Given the different capabilities of artificial and biological neural networks, it would be advantageous to build systems where the two types of networks are directly connected and can synergistically interact. As proof of principle, here we show how to create such a hybrid system and how it can be harnessed to improve animal performance on biologically relevant tasks. Using optogenetics, we interfaced the nervous system of the nematode *Caenorhabditis elegans* with a deep reinforcement learning agent, enabling the animal to navigate to targets and enhancing its natural ability to search for food. Agents adapted to strikingly different sites of neural integration and learned site-specific activation patterns to improve performance on a target-finding task. The combined animal and agent displayed cooperative computation between artificial and biological neural networks by generalizing target-finding to novel environments. This work constitutes an

initial demonstration of how to robustly improve task performance in animals using artificial intelligence interfaced with a living nervous system.

Artificial and biological neural networks differ in fundamental ways. Artificial neural networks can be trained to fit complicated functions using human-specified scoring metrics and have been used to accomplish a broad array of computational tasks[1]. However, artificial intelligence algorithms often fail to generalize, and may not perform well when applied to problems that are even slightly different from the ones on which they were trained[2]. Biological neural networks, on the other hand, have evolved to perform computations that help animals generalize to new and changing environments. The complementary strengths of artificial and biological neural networks raise the question of whether they can be integrated into a system that can not only compute information in a directed way but can also improve behavior while generalizing to novel situations.

Previous works have attempted to use direct neural stimulation to improve performance on a variety of tasks, relying on manual specification for stimulation frequencies, locations, dynamics, and patterns[3–6]. A central difficulty in this approach is that manual tuning is highly impractical, as activation patterns for a given task and set of neurons are often unknown[3] and there is a combinatorial explosion of stimulation parameters to test. In addition, effective patterns can vary depending on which neurons are targeted and on the animal itself[7,8]. Thus, even though technologies for precise neuronal modulation exist[9,10], there still lies the challenge of how an artificial intelligence algorithm can systematically and automatically learn strategies to activate a set of neurons to improve a particular behavior[11–15].

Here we addressed this challenge using deep reinforcement learning (RL), which can autonomously integrate with an animal's nervous system to improve behavior. In an RL setting, an agent collects rewards through interactions with its environment. By leveraging deep neural networks, RL algorithms have been able to successfully discover complex sequences of actions to solve a wide set of tasks[16–26]. These past successes relied on reward signals to train algorithms, a framework that can be readily adapted to biologically-relevant goals, such as finding food or mates. Consequently, an RL-based approach has the potential to handle the main computational problems in behavior improvement through neuronal stimulation.

To evaluate whether a deep RL agent can be trained to integrate with the nervous system by stimulating neurons to improve animal task performance, we interfaced an RL agent with the nervous system of the nematode *C. elegans* using optogenetic tools[9,12]. 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 behavior. The agents could not only successfully couple with different sets of neurons to perform a target-finding task, but could also generalize the task to improve food search across novel environments in a zero-shot fashion. This ability to generalize performance to novel environments is an important feature in natural behaviors and was achieved by augmenting the animal's native nervous system with artificial neural networks.

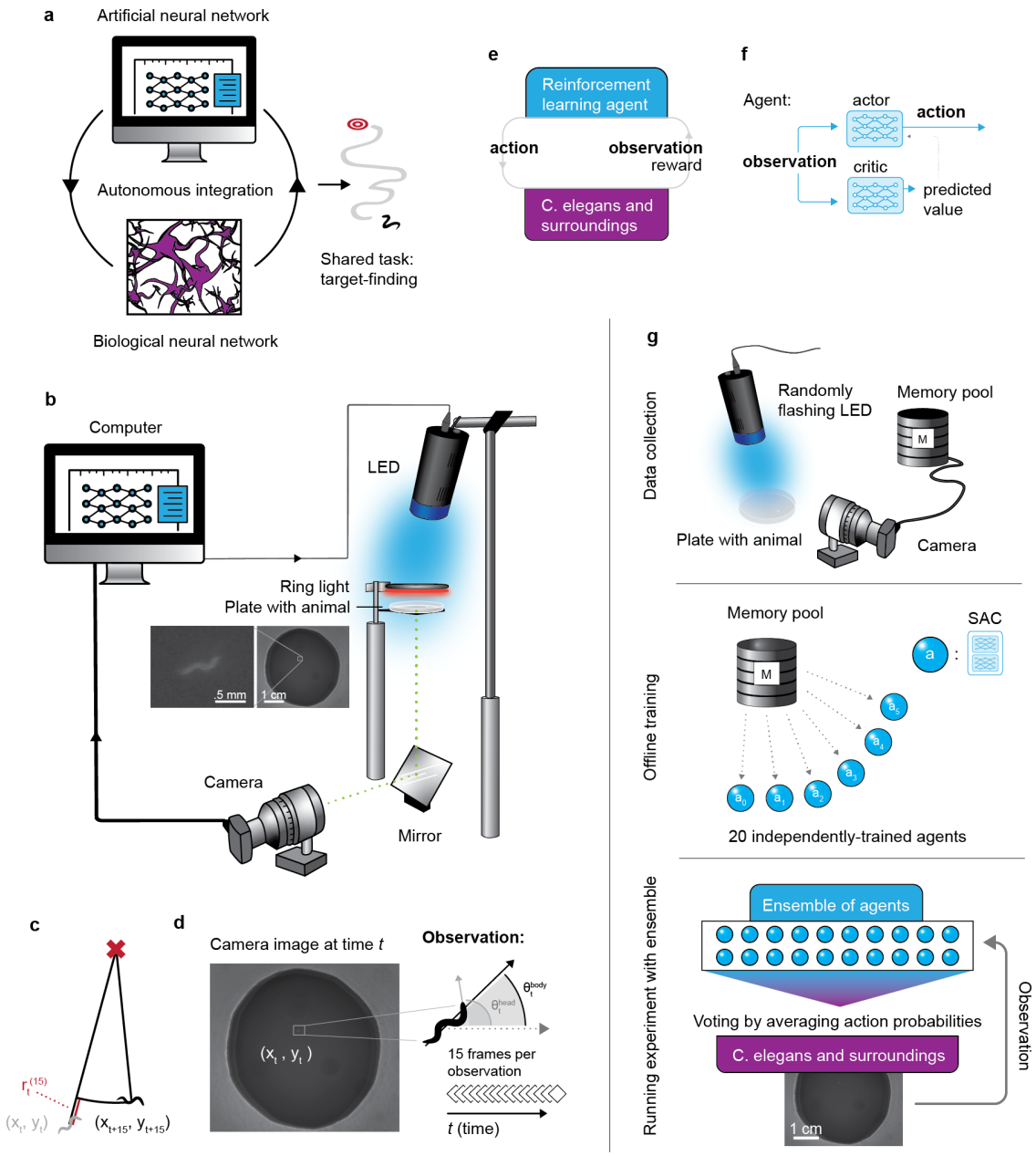

Fig. 1 | A system that integrates deep RL with the *C. elegans* neural network. a, Concept for combining artificial and biological neural networks for a shared task. **b,** Closed-loop setup using optogenetics. A single nematode was placed in a 4 cm-diameter field and illuminated by a red ring light 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 6 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 the coordinates of the animal's center of mass on the plate at time t (x_t, y_t), and the sines and cosines of the head and body angles, $(\theta_t^{body}, \theta_t^{head})$ 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. Animals were switched out approximately every 20 minutes. Twenty soft actor-critic agents were independently trained on the memory pool. During evaluation, the agents were put into an ensemble that voted in real time on actions. Each individual agent's decision was based on the observation received from the camera.

Name	Genotype	Expression
CH1	P*str-2::ChR2*	AWC(ON), [ASI]*
CH2	P*ttx-3::ChR2*	AIY
AR	P*npr-4::Arch*	SIA; SIB; RIC; AVA; RMD; AIY; AVK; BAG

* Bracketed neurons had weak or unstable expression in both our lines and the literature.

Table 1 | Transgenic line names in text with their genotypes and expression.

Connecting the nervous system to AI

We used a closed-loop setup to couple an RL agent to an animal's nervous system (Fig. 1a, b). We first formulated target-finding as an RL problem by defining a dense reward that increased with an animal's proximity to a 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. Features were automatically extracted from each camera frame to track the animal's center of mass (x_t, y_t) and its head and body angles $(\theta_t^{body}, \theta_t^{head})$ relative to the $+x$-axis. We took polar coordinates of the angle measurements so that for every frame at time t, we defined an observation $(\sin\theta_t^{body}, \cos\theta_t^{body}, \sin\theta_t^{head}, \cos\theta_t^{head}, x_t, y_t)$ (Fig. 1d). Each observation the agent received included these six variables from frames over the past five seconds, making agent inputs 90-dimensional (6 variables × 3 frames per second × 5 sec, Methods).

Given an observation at time t, the RL agent was trained to learn what action a_t to take at that time to maximize the return, defined as a sum of rewards discounted over time (Fig. 1e, Methods). To take an action, the agent could use optogenetics[9] to stimulate selected neurons that expressed channelrhodopsin, a light-gated ion channel that can be stimulated by blue light (480 nm) to

activate neurons[10]. An agent thus influenced animal behavior by deciding whether to turn an LED on or off at each timestep. As a first step, we used the transgenic line referred to as CH1 (Table 1), in which the *str2* promoter drives expression of channelrhodopsin in the sensory neuron AWC^{ON} (Fig. 2a). AWC^{ON} has been shown to activate when animals move away from attractive odors[27]. Consistent with this, an RL agent could flash blue light on a CH1 animal and cause it to turn around (Supplementary Video 1-2). It is important to note that prior to training, the RL agent had no built-in information about this turning action.

For the implementation of the RL agent, we chose the soft actor-critic (SAC) algorithm because of its successes in simulated and real-world RL environments[22,26,28,29]. SAC has separate neural networks for a critic that learns to evaluate observations and an actor that learns to optimize actions based on the critic evaluations and maximize return (Fig. 1f, Methods). Both neural networks take observations as input and consist of two layers with 64 units per layer (Methods). The actor outputs probabilities of turning the light on 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 a large amount of data for training. For instance, agents learning to play Atari can require thousands of hours of gameplay to achieve good performance[18,19]. It was infeasible to collect thousands of hours of recordings in our environment, and unlike videogames or physical systems with reliable dynamics, adequate computer simulations of the *C. elegans* nervous system and its behaviors are not available to generate training data[30]. Therefore, to facilitate algorithm development and reduce the amount of data needed to learn the target-finding task, agents were trained offline on pre-recorded data, which were collected for 20 min per animal

for a total of 5 h. During training data collection, the light was turned on with a probability of 0.1 every second (Fig. 1g, top and Methods). Following approaches in supervised learning[31], 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 in the target-finding task (Extended Data Fig. 1), similar to observations from previous work[32,33]. In simulated environments, such performance crashes can be quickly 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 the most effective for our environment (Extended Data, Fig. 2-5). The final deep RL agents were ensembles of 20 SAC agents, and the collection, training, and evaluation pipeline is shown in Fig. 1g.

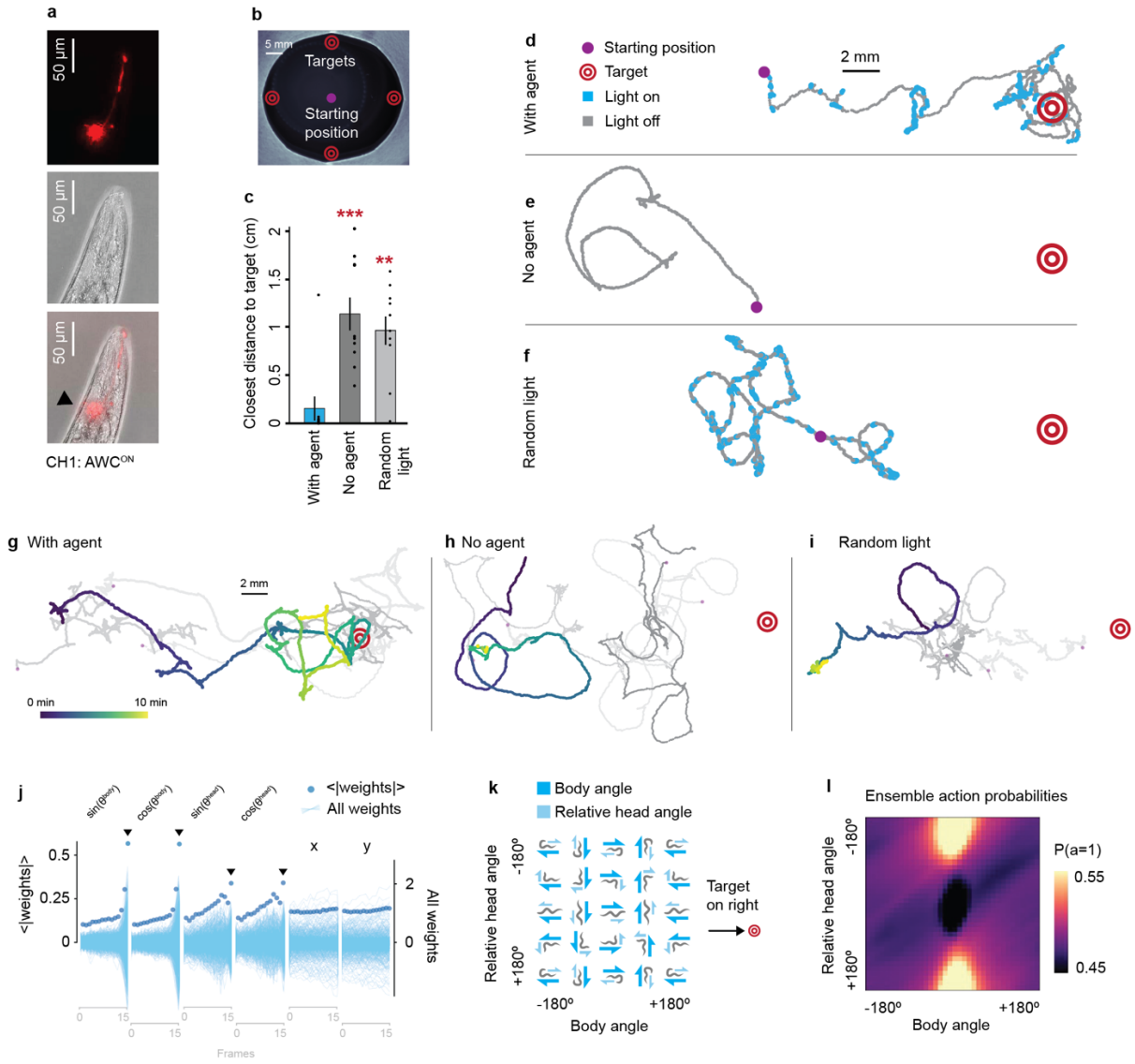

Fig. 2 | The system learned to navigate the *C. elegans* line CH1 to a target. a, Optogenetically modified neuron AWC^{ON} (black arrow) in the CH1 line. See Table 1 for transgenic line information. **b,** Evaluation setup. The animal was placed in the center (purple circle) of a filter paper circle 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,** 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. Error bars denote standard error. Mann-Whitney U Test, with agent vs. with control conditions indicated by asterisks, **P<.01, ***P<.001. **d-f,** Sample track with patterns of light activation along the trajectory (colored in blue) for animals with agent (**d**), without agent (**e**), and with randomly flashing light such that the total time with light on was the same as in 10 episodes of trials with agents (**f,** random light). With the agent, the animal moved to the target (red concentric circles) and stayed near it. Without agents, animals moved randomly. Purple dots denote starting location. **g-i,** Five sample tracks for each of the conditions in (**d-f**), with one arbitrarily chosen track colored by time. **j,** Weights of the first 64-neuron layer in all actor networks of the soft actor-critic ensemble. Weights for all neurons and all agents are plotted in light blue (axis on the right). Mean absolute values of weights are plotted in dark blue (axis on the left). For angle-related variables, the most recent frames (black arrows) have the largest weights. **k,** Reference for the agent action probability plot in **l**, showing example animal conformations arranged by body (x-axis) and head (y-axis) angles, that were sent as simulated inputs to agents. Input locations were fixed to the left of the target (see main text). **l,** Action probabilities (P(a=1), see color map on right) of the SAC ensemble trained on CH1 as a function of body relative to target location (x-axis) and head angles relative to body angle (y-axis).

Agents could navigate animals to targets

We first trained an agent on data collected on CH1 animals (Fig. 2a). To evaluate the agent, a single CH1 animal was placed in the center of a 4 cm-diameter arena and target coordinates were entered as an input to the agent. The agent was set to navigate the animal over a 10 min episode to a target placed in one of four possible locations (Fig. 2b). Figure 2d shows an example trace where the animal was navigated by the agent from a starting position towards a target. Upon reaching the target, the agent was also able to confine animals to the target area for the rest of the episode (Supplementary Video 2). In contrast, animals without an agent (Fig. 2e) and animals with random light intervention (Fig. 2f, Supplementary Video 1) were unable to reach targets. The trained agent could consistently navigate animals to targets better than no agent and random light conditions (Fig. 2c, p=.0005, no agent; p<.003, random light; Mann-Whitney U Test, n=10, Fig 2g-i), showing that the RL agent successfully coupled with CH1 animals and learned a target-finding strategy.

To understand what the agent trained on CH1 learned, 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 to assess which input variables were associated with large weights (Fig. 2j). Measurements of head and body angles corresponding to the most recent frame in an observation (black arrows in Fig. 2j) had larger weight magnitudes than ones from earlier frames. Therefore, to visualize agent strategies, we fixed the values of the 30 coordinate variables ($(x_{t'}, y_{t'});\ t - 5\ s < t' < t$) in each observation to a position left of the target (Methods) and plotted the probability that the ensemble turned the light on as a function of body and head angles at the latest time in the observation ($\theta_t{}^{body}, \theta_t^{head}$) (Fig. 2l).

157

158 For example, the animal posture at $\theta_t^{body} = 0°$ and $\theta_t^{head} = 0°$ in the center of Fig. 2k corresponds
159 to the center of Fig. 2l where the agent learned that $P(a_t = 1) < 0.5$. This means that when the
160 animal's body and head were pointed at 0° toward the target, the agent learned to turn the light
161 off. In contrast, the observations where the agent was most likely to turn the light on and activate
162 AWC^{ON} were ones where the animal's body was pointed toward the target but the head was turned
163 away. These visualizations along with the agent's success during evaluations demonstrated that by
164 probing deep RL agents trained on this task, we could learn about patterns of neural activations
165 that could produce a desired behavior.

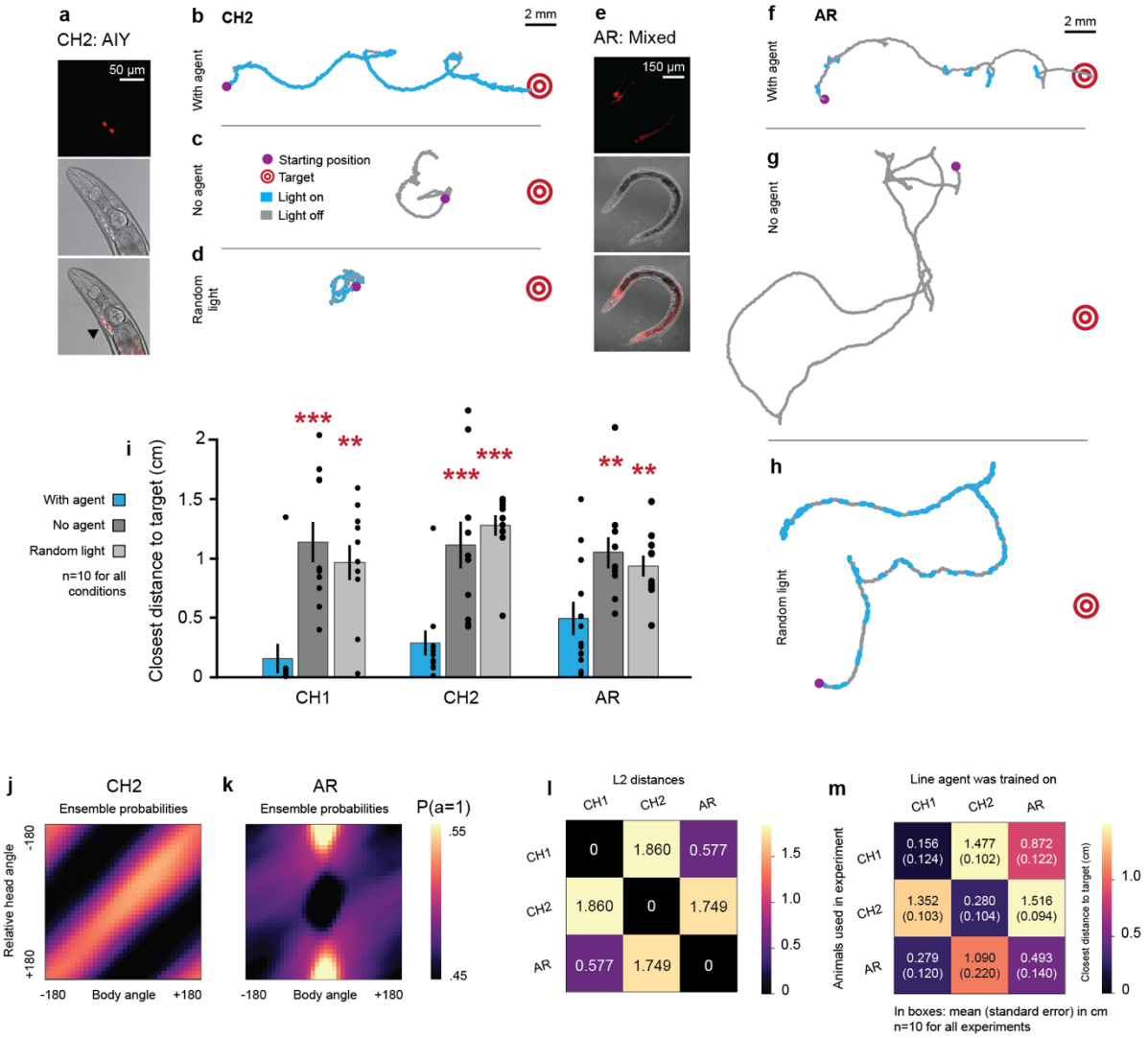

Fig. 3 | The system learned to navigate different optogenetic lines to a target with neuron-specific strategies. a, Optogenetically modified interneuron AIY in the CH2 line (Table 1). **b,** Following the format in Fig. 2d-f, example tracks with positions of light activation along the trajectory highlighted in blue for animals with the agent, **c,** without any optogenetic activation, and **d,** with randomly flashing light. In **b-d**, **f-h**, 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. Even though the animals started closer to the target in the two control conditions, they still did not reach the target. **e,** Optogenetically modified interneurons, sensory neurons, and motor neurons in the AR line (Table 1). **f,** Example tracks with light activation for animals with agent, **g,** without optogenetic activation, **h,** with randomly flashing light, again with locations along the trajectory of light on in blue. **i,** Following Fig. 2c closest distances to target achieved by each genetic line with agent, no agent, and random light. Animals with agents were significantly more successful in target navigation than animals without agents. Mann-Whitney U Test, control condition vs. with agent condition indicated by asterisks, **P<.01, ***P<.001. For CH2, p<.0006, no agent; p<.0002, random light. For AR, p<.007, no agent; p<.008, random light. The first three bars in this figure are reproduced from Fig. 2c for comparison purposes. **j,** Action probabilities of SAC agents trained on line CH2, plotted in coordinates from Fig. 2k. **k,** Action probabilities of agents trained on AR. **l,** L2 distances between ensemble action probability matrices for each genetic line. **m,** Agents trained on the three genetic lines CH2, CH1, and AR were tested on each of the other lines *without retraining*. The 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 (**l**) correlate with the closest distances achieved in across-policy evaluation experiments (**m**) (r^2=.8578, p <.0004).

166 **The agent adapted to different neurons**

167 We aimed to build a robust and flexible algorithm that could be trained to adapt to its connected
168 neurons, asking whether the RL agent could learn appropriate rules for a variety of neural
169 connections without any explicit prior knowledge about them. We therefore tested our approach
170 on transgenic lines that were functionally distinct from CH1. First, we tested a line referred to here
171 as CH2, which expresses channelrhodopsin specifically in AIY interneurons (using the *ttx-3*
172 promoter, Table 1, Fig. 3a). AIY neurons are involved in chemotaxis[11] and suppress turning,
173 whereas AWC^{ON} (the modified neuron in CH1) causes turning. When an agent was trained on CH2
174 and evaluated as in Fig. 2b-d, the agent successfully navigated an animal to a target (Fig. 3b) while
175 control animals did not reach targets (animal without agent in Fig. 3c and with random light in Fig.
176 3d). Again, the agent achieved this consistently better than no agent and random light conditions
177 (Mann-Whitney U Test, $p<.0006$, no agent; $p<.0002$, random light); see Fig. 3i, center,
178 Supplementary Videos 3 (random light control) and 4 (with agent), and Extended Data Fig. 6a-c.
179
180 In the cases considered so far, agents interacted with a single neuron type in the animal. We next
181 asked whether our approach would work when an agent modulated the activity in multiple neuron
182 types instead of one. To this end, we used the line AR, which is expressed in many neuron types
183 (using the *npr-4* promoter, see Table 1, Fig. 3e). Unlike previous genetic lines which expressed
184 channelrhodopsin, AR animals expressed archaerhodopsin, which inhibits neurons upon
185 stimulation with green light (540 nm). This line tested the abilities of the RL agent with a different
186 set of neuronal connections and a different means of neural modulation. Animals with the trained
187 agent once again moved closer to targets than control animals (Fig. 3f-h; statistics in Fig. 3i, right;
188 see Supplementary Videos 5-6 and Extended Data Fig. 6d-f for additional examples). It is

interesting to note that there was no previously characterized behavioral phenotype for optogenetic activation of this line (see Bhardwaj et al. for *npr-4* mutant behavior), yet the agent still learned to direct these animals towards a target.

Agents predicted similarities between neural circuits

To confirm that agents learned action probabilities tailored to their respective neural connections, we plotted agent action probabilities in Fig. 3j-k in the 2-dimensional subspace of θ_t^{body} and θ_t^{head} as in Fig. 2k (Extended Data Fig. 7). Although the behavior of CH1 in response to blue light is mostly to reverse and CH2 is mostly to move forward, agent policies were not merely inverses of each other. Rather, agents learned that CH2 control was dependent largely on the animal's head angle relative to the target while CH1 and AR control depended on specific head and body angle combinations. Despite large differences in the CH1 and AR lines (excitation of a single neuron in CH1 versus inhibition of multiple neurons in AR), training on AR resulted in an action probability matrix that was strikingly similar to the one from training on CH1. To quantify these similarities in learned actions for the different lines, we measured L2 norm differences of the action probability matrices (Fig. 3l). To assess how well this metric for agent differences corresponded to differences in animal behavior, we performed cross-evaluation experiments using the target navigation task in Fig. 2b and tested the agent for each line on animals from each of the other lines (Fig. 3m).

The matrix of cross-evaluation results in Fig. 3m correlated well with predictions based on the similarity of the action probability matrices in Fig. 3l (r^2=.8578, p <.0004). As expected from the contrast in action probabilities in Fig. 3j (CH2) versus Fig. 2l (CH1) and 3k (AR), CH2 did not respond well to agents trained on CH1 or AR. For example, when the agent trained on the CH2

line was tested with an animal from the CH1 line, the closest distance reached from the target was about 1.477±0.102 cm, much larger than when tested on the same CH2 line, 0.280±0.104 cm (Fig. 3m). The closest distance was also comparable to or greater than the no agent or random light conditions for CH1 (Fig. 3i), as the CH2 agent tended to drive CH1 animals away from rather than toward targets (p-value<.08, no agent; p-value<.009, random light; Mann-Whitney U Test). Likewise, neither CH1 nor AR animals performed well on the task when paired with the CH2 agent.

Surprisingly, we also found that both CH1 and AR lines were most successful when paired with the CH1 agent even though the AR agent was trained on data from the line itself (p<.002, CH1 line with CH1 vs. AR agent; p<.04, AR line with CH1 vs. AR agent, Mann-Whitney U Test, n=10). These results may be explained by higher data quality caused by the stronger response of CH1 to optogenetic stimulation (Supplementary Videos 1, 2, 5, 6), reflected in the greater action certainties in the CH1 ensemble as compared to the AR ensemble (Fig. 2l, 3k). 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 behavior of these lines under optogenetic control in the target-finding task.

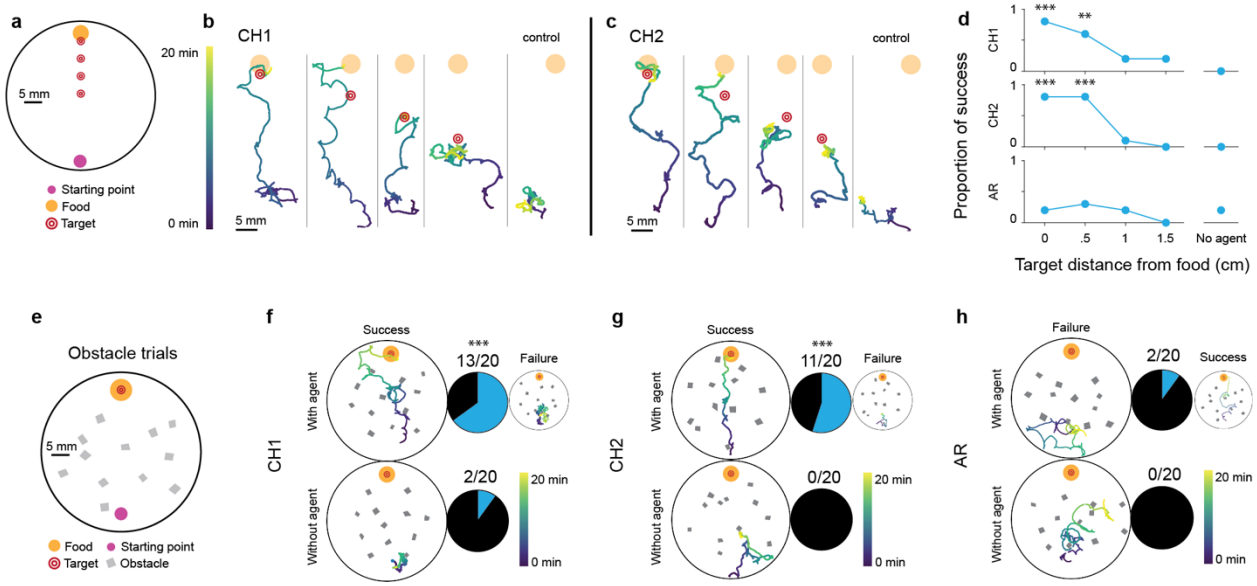

Fig. 4 | Agents generalize to novel situations by performing computations that cooperate with the *C. elegans* nervous system. a, Diagram of error-handling food search experiments. A single animal was placed at the opposite end of a plate (starting location large purple circle) as a 5 μm drop of OP50 *E. coli* bacteria (orange circle). Trials lasted 20 min each and success was defined by whether the animal reached food. Agents were directed to navigate animals to a target a distance away from the food (agent target location denoted by concentric red circles). **b,** Sample tracks for CH1 animals with agent that either succeeded (columns 1, 2) or failed (columns 3, 4) to reach food, based on the majority result of trials with the target at the given distance from the food. A control track without an agent is shown in the fifth column. **c,** Sample tracks for CH2 animals as in **b**. **d,** Proportion of animals that successfully reached food for CH1, CH2, and AR, plotted as a function of the target distance from food. Data are also shown for trials with no agent (n=10 for every experimental condition) For CH1 and CH2, targets up to 0.5 cm away led to significantly better performance than without agents. **P<.01, ***P<.001 (with agent vs. no agent; p<.0004 for CH1 with target at 0 cm from food and CH2 with target at 0 and 0.5 cm from food; p<.006 for CH1 with target at 0.5 cm from food). Results were not statistically significant for line AR. **e,** A diagram of the plate used for experiments with obstacles. Twelve paper rectangles with side lengths approximately 2 mm were scattered between the animal and food. For each trial a single animal was placed on a plate at the opposite end (animal's starting point denoted by purple circle) of a 5 μm drop of food (OP50 *E. coli* bacteria). Trials lasted 20 min and success was defined by whether the animal reached food. Agents were directed to navigate animals to the food. **f,** Sample tracks for CH1 animals that successfully reached food with the agent (top left), failed to make it to food with the agent (top right), and a control trial without the agent (bottom). Success rates shown in blue and black pie charts. 13/20 animals succeeded with the agent and 2/20 without. Animals with agents were significantly more likely to make it to food than animals without agents; ***P<.001 (permutation test, p<.0004). **g,** Sample tracks for CH2 animals. 11/20 animals reached food with the agent and 0/20 animals without (permutation test, p<.0001). **h,** Sample tracks for AR animals, with a failed trial in the top left to represent the majority outcome. 2/20 animals reached food with the agent and 0/20 without (permutation test, p=.244).

Agents cooperated with nervous systems for food search

We next evaluated whether agents and animals could transfer their abilities from the target-finding task to improve food search. We tested two scenarios: first, whether the animal could correct errors made by an agent about the location of food, and second, whether the animal and agent could navigate an unforeseen environment with obstacles to reach food. Both scenarios represented novel environmental conditions, and because agents were not retrained in either case, they needed to show evidence that when interfaced with the animal, the combined system could generalize target-finding to the food search task. Both tasks also required the animal to contribute information from its sensory system to find food, so the experiments tested cooperativity between artificial and biological neural networks beyond the previous target-finding experiments.

For the error-handling task, targets were placed at increasing distances from the edge of a 5 μL patch of food (OP50 *E. coli* bacteria) to mimic errors made by the agent (Fig. 4a; Methods). Agents were on throughout the experiment; crucially, they were not switched off when animals reached the target. Animals were tested on whether they could reach the food in 20 min trials with or without RL agents. Agents were identical to the ones used in Fig. 2 and 3. For both CH1 and CH2 lines, when targets were 0.5 cm away from food edges, animals were able to leave an agent's target region (a circle of radius 0.0625 cm; Methods) and moved to the food in 8/10 trials ($p<.0004$). This was significantly different from trials without any agent assistance (Fig. 4b-c, "no agent"), in which 0 animals reached food in 10 trials for both CH1 and CH2 lines. AR was not as successful with agent assistance (Fig. 4d, bottom; Extended Data Fig. 8), likely due to the less reliable control in moving animals to a target. This suggests that simultaneous modulation of the neurons in this line is not as strongly linked to directed movement as in lines CH1 and CH2 (Fig. 3i, right). In

252 contrast, CH1 and CH2 animals could effectively switch between making decisions based on their
253 own sensory systems or the agents, which were trained to keep animals at targets.
254
255 We then designed a trial in which twelve paper quadrilaterals with 1-3 mm edges (comparable to
256 the 1 mm body length of *C. elegans*) were scattered randomly on the plate to serve as obstacles
257 between an animal and a 5 μL patch of food (Fig. 4e; Methods). In this scenario, animals were
258 again tested on whether they could reach food during a 20 min trial with and without agents. This
259 was a particularly challenging task because animals had to use their sensory and motor systems to
260 navigate around obstacles, while agents had to navigate animals to food despite noisy movements
261 caused by obstacles. CH1 and CH2 animals performed very well in navigating this new
262 environment to find food (Fig. 4f-g, p-value<.0004, CH1; p-value<.0001, CH2; permutation tests).
263 The AR line was not as successful (Fig. 4h); overall, the agent could navigate AR animals closer
264 to targets but could not achieve more difficult food search tasks. For CH1 and CH2, however, these
265 data provide evidence that our system displays cooperative computation between artificial and
266 biological neural networks to improve *C. elegans* food search in a zero-shot fashion without any
267 retraining in novel environments.
268
269 **Discussion**
270 We showed here how to build a hybrid system where deep RL can interact with an animal's
271 nervous system to improve a target behavior. In the data-limited context of biological systems, we
272 could train deep RL agents using data augmentation and improve the stability of deep RL using an
273 ensemble of agents. Agents could customize themselves to specific and diverse sites of neural
274 integration. These results did not depend on the number of neurons that agents were interfaced

with, nor whether the interactions were excitatory or inhibitory. In addition, the animal plus agent system could generalize a learned target-finding strategy to novel environments for food search. We demonstrated that the inherent ability of the *C. elegans* nervous system to find food could be enhanced by deep RL, helping animals find targets faster and in more challenging environments than they could on their own.

In previous work, brain-machine interfaces have allowed animals to control machines through neural recordings[34–36]. Conversely, supervised optogenetic manipulations have taken control of *C. elegans* neurons or muscles to turn the animal into a passive robot[11,37]. 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 activation patterns to interact with the nervous system, and did so in a way that allowed computations from both networks to drive animal behavior in a robust manner that generalizes in a zero-shot fashion to novel environments. Our system was also able to discover patterns of neural activity that were sufficient to drive specific behaviors: studies of sufficiency complement the more traditional lesion and inhibition studies in neuroscience, which have historically only focused on determining the neural circuitry correlated with or necessary for specific behaviors.

We used *C. elegans* as a model organism for its small and accessible nervous system. It would be interesting for future work to test our method in larger state spaces and action spaces, as one would find in an animal with a richer behavioral repertoire and larger nervous system. Deep RL has already solved complex simulated tasks in high dimensional spaces with large numbers of parameters[16,18,20], suggesting its potential for integration with larger animals. Overall, our study

298 opens new avenues for understanding neural circuits, improving behavior using deep RL, and
299 building hybrids between artificial and biological networks that can utilize the flexibility,
300 robustness and computational power of AI and animals.

Methods

Animal genetics and care

Genetic lines.

Strains are listed in Extended Data Table 1. All animals had *lite-1* mutant backgrounds to reduce light sensitivity.

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 Figure 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 110mm 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 Figure 4 were not soaked in copper solution. If food patches were used in the experiment as in Figure 4, 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 1s. 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.

Reinforcement learning details

Reinforcement learning (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 soft actor-critic (SAC) algorithm for all agents[26,28]. 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* behavior[11]. We used pixel coordinates (x, y) of the animal's centroid location in the image, the body angle relative to the $+x$-axis and the head angle relative to the $+x$-axis (see Fig. 1). Body angles were computed by fitting a line to a skeletonized worm image and head angles were computed through template matching. See the code in `improc_v.py` for details.

Head/tail identification was done by assigning the head label to the endpoint that was closest to the head endpoint in a previous frame. To handle reversals, a common behavior 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. 15 frames (5 s at 3 fps) were concatenated together for a single observation. Coordinates were normalized so their means in each 15-frame observation was within $[-0.5, 0.5]$. An observation $\boldsymbol{s_t}$ corresponding to time t was thus comprised of $6 \times 15 = 90$ variables:

$$\boldsymbol{f_t} = (sin\theta_t^{body}, sin\theta_t^{body}, sin\theta_t^{body}, sin\theta_t^{body}, x_t, y_t)$$

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

Above, $\boldsymbol{f}_t$ 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 (see Methods: Standard evaluation).

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

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

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 reinforcement learning, 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,

$$\sum_t \mathbb{E}_{(\boldsymbol{s_t}, \boldsymbol{a_t}) \sim \rho_\pi} [\gamma^t r_t(\boldsymbol{s_t}, a_t)].$$

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:

$$\sum_t \mathbb{E}_{(\boldsymbol{s_t}, a_t)\sim\rho_\pi}\left[\gamma^t r_t(\boldsymbol{s_t}, a_t) + \alpha\mathcal{H}(\pi(\cdot\,|\boldsymbol{s_t}))\right].$$

The sum now contains an added entropy term $\mathcal{H}$ of the policy $\pi(\cdot\,|\boldsymbol{s_t})$, scaled by a temperature parameter α. $\pi(\cdot\,|\boldsymbol{s_t})$ signifies the policy function π over all possible events. We used a discrete version of SAC with automatic entropy tuning (see code for implementation).

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

$$\boldsymbol{m_t} = (\boldsymbol{s_t}, a_t, r_t, \boldsymbol{s_{t+15}})$$

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 centered such that the average of the location coordinates were at the origin, (0,0) pixels. Then a location within a ±450-pixel square (comparable to the size of the evaluation arena) was drawn from a uniform distribution and the coordinates recentered 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 Extended Data Table 2 for architecture and hyperparameter choices. 20 agents per genetic line were trained independently on the same memory buffer for 20 epochs of 5000 steps each. Minibatch size 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 Extended Data Fig. 2-4).

Independent agents were trained such that the randomly taken action a_t, reward r_t, and the associated states $\boldsymbol{s_t}$ and $\boldsymbol{s_{t+15}}$ 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 Haarnoja et al.[26] 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,

$$\pi_{ensemble}(\boldsymbol{a_t}|\boldsymbol{s_t}) = \frac{1}{N}\sum_{n}^{N}\pi_n(a_t|\boldsymbol{s_t})$$

where $N = 20$. If the average probability $\pi_{ensemble}(a_t|\boldsymbol{s_t}) \geq 0.5$, then the light was on at that timestep.

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 GPUs 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. 2j, Extended Data Fig. 7), 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 $(x_{t\prime}, y_{t\prime})$; $t - 5 < t' < t$) were always fixed to 0.9375 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 1296 different input observations, each of which were 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 4cm-diameter filter paper barrier soaked in copper (II) chloride. All animals were cultured on food with ATR and were thus sensitive to optogenetic perturbation.

Standard evaluation.

Animals were placed in the center of the field. A target was randomly chosen among top, bottom, left, and right options (see Fig. 2b). The experiment 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 0.2896 for CH1, 0.4647 for CH2, and 0.3844 for AR. Animals were recorded while light decisions were made every 1 s, with the probability of light on according to the genetic lines listed.

Cross-agent evaluation.

For the plot in Figure 3m, 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. 10 trials of 10 min each were performed for every agent-genetic line combination.

Error-handling food search experiments.

For the food search experiments in Figure 4a-d, a 10 cm NGM plate was prepared with a 4 cm-diameter filter paper circle soaked in 20 mM copper (II) chloride. 5 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 cm, 0.5 cm, 1 cm, or 1.5 cm away 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 1 CH1 animal left food after reaching it during a food search trial when the target was placed 1 cm away from the food edge. This trial was counted as a success.

Obstacle food search experiments.

For the obstacle trials in Figure 4e-h, 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. 4e-h. Plates were replaced with new obstacle arrangements every 5-10 trials. 5 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 Figure 4e, with the food droplet on the other end and the obstacles in 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.

Data and code availability

Processed animal tracks, analysis code, and training code examples are available at https://tinyurl.com/RLWorms. Other data are available upon request.

Author Contributions

CL, GK, and SR designed the study. CL wrote code, performed experiments, and did data analysis. CL, GK, and SR wrote the manuscript.

Acknowledgments

We thank Surya Bhupatiraju for discussions about reinforcement learning and comments on the manuscript. We thank Timothy Hallacy and Abdullah Yonar for guidance in *C. elegans* experiments and Cory McCartan for input on statistical analyses. We thank Kenneth Blum, Cengiz Pehlevan, Giri Anand, Alexandru Bacanu, Benjamin Brissette, Dianna Hidalgo, Roya Huang, Heitor Megale, William Weiter, Yusuf Ilker Yaman, Vincent Zhuang, and Steven Zwick for comments on the manuscript.

This work was supported in part by NIGMS grant 1R01NS117908-01 (SR), Dean's Competitive Fund from Harvard University (SR, CL), NIH R01EY026025 (GK), and an NSF GFRP fellowship (CL).

Competing interests

The authors declare no competing interests.

505 **References**
506
507 1. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
508 2. Sinz, F. H., Pitkow, X., Reimer, J., Bethge, M. & Tolias, A. S. Engineering a Less Artificial
509 Intelligence. *Neuron* **103**, 967–979 (2019).
510 3. Iturrate, I., Pereira, M. & Millán, J. del R. Closed-loop electrical neurostimulation: Challenges
511 and opportunities. *Curr. Opin. Biomed. Eng.* **8**, 28–37 (2018).
512 4. Xu, J. *et al*. Thalamic Stimulation Improves Postictal Cortical Arousal and Behavior. *J.*
513 *Neurosci.* **40**, 7343–7354 (2020).
514 5. Bonizzato, M. & Martinez, M. An intracortical neuroprosthesis immediately alleviates
515 walking deficits and improves recovery of leg control after spinal cord injury. *Sci. Transl.*
516 *Med.* **13**, eabb4422 (2021).
517 6. Enriquez-Geppert, S., Huster, R. J. & Herrmann, C. S. Boosting brain functions: Improving
518 executive functions with behavioral training, neurostimulation, and neurofeedback. *Int. J.*
519 *Psychophysiol.* **88**, 1–16 (2013).
520 7. Bergmann, E., Gofman, X., Kavushansky, A. & Kahn, I. Individual variability in functional
521 connectivity architecture of the mouse brain. *Commun. Biol.* **3**, 1–10 (2020).
522 8. Mueller, S. *et al*. Individual Variability in Functional Connectivity Architecture of the Human
523 Brain. *Neuron* **77**, 586–595 (2013).
524 9. Husson, S. J., Gottschalk, A. & Leifer, A. M. Optogenetic manipulation of neural activity in
525 C. elegans: from synapse to circuits and behaviour. *Biol. Cell* **105**, 235–250 (2013).
526 10. Nagel, G. *et al*. Channelrhodopsin-2, a directly light-gated cation-selective membrane
527 channel. *PNAS* **100**, 13940–13945 (2003).

11. 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).

12. 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).

13. Wen, Q. *et al*. Proprioceptive Coupling within Motor Neurons Drives C. elegans Forward Locomotion. *Neuron* **76**, 750–761 (2012).

14. Hernandez-Nunez, L. *et al*. Reverse-correlation analysis of navigation dynamics in Drosophila larva using optogenetics. *eLife* **4**, e06225 (2015).

15. Donnelly, J. L. *et al*. Monoaminergic Orchestration of Motor Programs in a Complex C. elegans Behavior. *PLOS Biol*. **11**, (2013).

16. Silver, D. *et al*. Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484–489 (2016).

17. Silver, D. *et al*. Mastering the game of Go without human knowledge. *Nature* **550**, 354–359 (2017).

18. Schrittwieser, J. *et al*. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature* **588**, 604–609 (2020).

19. Mnih, V. *et al*. Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).

20. Vinyals, O. *et al*. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* **575**, 350–354 (2019).

21. OpenAI *et al*. *Dota 2 with Large Scale Deep Reinforcement Learning*. http://arxiv.org/abs/1912.06680 (2019) doi:10.48550/arXiv.1912.06680.

551 22. Wurman, P. R. *et al*. Outracing champion Gran Turismo drivers with deep reinforcement
552 learning. *Nature* **602**, 223–228 (2022).
553 23. Degrave, J. *et al*. Magnetic control of tokamak plasmas through deep reinforcement
554 learning. *Nature* **602**, 414–419 (2022).
555 24. Ibarz, J. *et al*. How to train your robot with deep reinforcement learning: lessons we have
556 learned. *Int. J. Robot. Res.* **40**, 698–721 (2021).
557 25. Haydari, A. & Yılmaz, Y. Deep Reinforcement Learning for Intelligent Transportation
558 Systems: A Survey. *IEEE Trans. Intell. Transp. Syst.* **23**, 11–32 (2022).
559 26. Haarnoja, T. *et al*. Soft actor-critic algorithms and applications. *ArXiv Prepr.*
560 *ArXiv181205905* (2018).
561 27. Chalasani, S. H. *et al*. Dissecting a circuit for olfactory behaviour in Caenorhabditis
562 elegans. *Nature* **450**, 63–70 (2007).
563 28. Christodoulou, P. Soft actor-critic for discrete action settings. *ArXiv Prepr.*
564 *ArXiv191007207* (2019).
565 29. Wong, C.-C., Chien, S.-Y., Feng, H.-M. & Aoyama, H. Motion Planning for Dual-Arm
566 Robot Based on Soft Actor-Critic. *IEEE Access* **9**, 26871–26885 (2021).
567 30. Sarma, G. P. *et al*. OpenWorm: overview and recent advances in integrative biological
568 simulation of Caenorhabditis elegans. *Philos. Trans. R. Soc. B Biol. Sci.* **373**, 20170382
569 (2018).
570 31. Shorten, C. & Khoshgoftaar, T. M. A survey on Image Data Augmentation for Deep
571 Learning. *J. Big Data* **6**, 60 (2019).
572 32. Reinforcement Learning Resources — Stable Baselines 2.10.2 documentation.
573 https://stable-baselines.readthedocs.io/en/master/guide/rl.html.

574 33. Nikishin, E. *et al*. Improving Stability in Deep Reinforcement Learning with Weight
575 Averaging. 5.
576 34. Andersen, R. A., Aflalo, T., Bashford, L., Bjånes, D. & Kellis, S. Exploring Cognition
577 with Brain–Machine Interfaces. *Annu. Rev. Psychol.* **73**, 131–158 (2022).
578 35. Tankus, A., Fried, I. & Shoham, S. Cognitive-motor brain–machine interfaces. *J. Physiol.*
579 *Paris* **108**, 38–44 (2014).
580 36. Sussillo, D., Stavisky, S. D., Kao, J. C., Ryu, S. I. & Shenoy, K. V. Making brain–
581 machine interfaces robust to future neural variability. *Nat. Commun.* **7**, 1–13 (2016).
582 37. Dong, X. *et al*. Toward a living soft microrobot through optogenetic locomotion control
583 of Caenorhabditis elegans. *Sci. Robot.* **6**, (2021).
584 38. Tandon, P. pytorch-soft-actor-critic. https://github.com/pranz24/pytorch-soft-actor-critic
585 (2022).
586 39. alirezakazemipour/Discrete-SAC-PyTorch: PyTorch implementation of discrete version
587 of Soft Actor-Critic. https://github.com/alirezakazemipour/Discrete-SAC-PyTorch.
588