Delayed Gratification in a Robotic Model of Cuttlefish Behavior

Qian Ying Wong[1], Kary Zheng[2], Shana Chao[3], Vansh Chhabra[2], Nange Hai[1], Alina Yuan[4], Yingshu Zhu[4], Jeffrey L. Krichmar[1,2(✉)], and Harrison Espino[2]

[1] Department of Cognitive Sciences, University of California, Irvine, CA 92697, USA
{qywong,xhai2,jkrichma}@uci.edu
[2] Department of Computer Science, University of California, Irvine, CA 92697, USA
{zhengj29,vchhabr1,espinoh}@uci.edu
[3] Department of Computer Engineering, University of California, Irvine, CA 92697, USA
shanasc@uci.edu
[4] Department of Art, University of California, Irvine, CA 92697, USA
{yuanax,yingsz3}@uci.edu

Abstract. Delayed gratification requires an agent to forgo an immediately available lower-value reward in order to obtain a higher-value reward after a delay. Inspired by cuttlefish behavior, we developed a model of delayed gratification and embodied it in a neurorobotic task. Following experiments with cuttlefish, the model learned to select between live shrimp, dead shrimp, or unobtainable prey options. A Q-learning agent learned reward identity, while a delay-dependent patience function selectively discounted the live-shrimp value during decision-making. The agent learned to prefer live shrimp more than dead shrimp and was willing to wait for the preferred prey. In a control condition, performance remained comparatively stable across delays. The policy was evaluated on a TurtleBot4 Lite robot through both simulation-to-robot transfer and on-robot training with color-based cues in a physical marine-inspired arena. Together, these results show that delayed-gratification-like behavior can be reproduced with a compact reinforcement-learning mechanism and partially embodied in a mobile robot. It suggests a parsimonious model for supporting delayed gratification in cephalopods and other organisms.

Keywords: Cephalopods · Cognition · Delayed gratification · Neurorobotics · Reinforcement learning

1 Introduction

Delayed gratification, which is the ability to forgo a lower-value, immediately available reward in order to obtain a higher-value one at a later time, is a basic ingredient of self-control. Although it is classically studied in humans, the same problem appears in computational form whenever a learning agent must trade

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Brock et al. (Eds.): SAB 2026, LNAI 16965, pp. 153–164, 2027.
https://doi.org/10.1007/978-3-032-39953-3_13

off short-term against long-term value, as is standard in reinforcement learning [14]. Delayed gratification becomes relevant to animal cognition and to artificial agents that must make trade-offs between immediate and delayed rewards.

Recent work has shown that common cuttlefish (*Sepia officinalis*), an invertebrate, can wait for a preferred but delayed prey item and refrain from waiting when waiting yields no benefit [11]. This behavior suggests that flexible reward-based decision-making is not limited to vertebrates and may arise from relatively compact computational mechanisms. Our group has previously taken this approach when modeling cuttlefish episodic-like memory [5], and the same logic motivates the present work. Because simulations can test whether a particular learning rule is sufficient to reproduce this behavior, and robots can test whether the policy remains functional under embodied sensing and action, delayed gratification is a useful target for neurorobotic modeling [4,6,7]. Our goal here is to test whether a biologically motivated controller can reproduce the central behavioral signature of delayed gratification in simulation and support similar behavior on an embodied robot.

To address this, we built a four-state foraging task adapted from the cuttlefish protocol of Schnell et al. [11]. Two states are experimental, pitting an immediately available low-value option (a dead shrimp) against a delayed high-value option (a live shrimp); the other two are control states, where the high-value option is unobtainable. A Q-learning agent [14] learns the reward value of each option, and a delay-dependent patience function, adapted from the patience model of Xing et al. [15], modulates the agent's willingness to wait. The patience term is applied selectively to the live-shrimp value at decision time, leaving the dead-shrimp and unobtainable values untouched, so that delay discounting is dissociated from the underlying learning of reward identity. The contributions of this paper are the following:

- We formalize a cuttlefish-inspired delayed-gratification task as a reinforcement-learning problem.
- We show that Q-learning combined with a selectively applied patience function is sufficient to reproduce the experiments that characterize delayed gratification in cuttlefish.
- We demonstrate that the learned policy can be embodied on a TurtleBot4 robot using color-based perception and navigation, completing the perception–decision–action loop on real hardware.

2 Related Work

2.1 Delayed Gratification in Humans and Animals

The behavioral study of delayed gratification began with Mischel and Ebbesen's marshmallow experiment [8]. For our purposes, a more direct paradigm was a study by Schnell and colleagues, who tested delay of gratification in the common cuttlefish, *Sepia officinalis* [11]. Cuttlefish were placed between two transparent chambers. One chamber contained an immediately available but less preferred

prey item; the other contained a more preferred prey that became accessible only after a delay. Crucially, eating the immediate option canceled the delayed option, so the cuttlefish faced a real trade-off between an easy meal and a better one later. Cuttlefish reliably waited for the preferred prey, in some cases for more than two minutes. In control conditions where waiting yielded no benefit, the same animals did not wait. This experimental–control dissociation, rather than waiting alone, is the behavioral signature that we target with our model. Because cuttlefish are invertebrates with a nervous system organized very differently from that of mammals, the result suggests that the underlying computation is not unique to vertebrates and may be an important aspect of cognition across species.

2.2 Reinforcement Learning and the Neuroscience

Reinforcement learning (RL) provides the standard computational language for delayed reward [14]. An RL agent learns by trial and error, updating estimates of the long-term value of each action so as to maximize cumulative reward. Q-learning is the canonical model-free instance of this idea: the agent maintains a table or function approximator over state–action values and updates each entry using the difference between the reward received and the value expected.

The relationship between RL and biology has been bidirectional from the beginning. Early RL algorithms, including the Rescorla–Wagner rule, were inspired by classical animal-conditioning experiments [13]. The flow then reversed: temporal-difference learning, originally developed as a tool for artificial agents, has become the standard framework for interpreting phasic dopamine activity, with dopaminergic responses now understood as a neural correlate of reward prediction error [13]. Hassabis et al. argue more broadly that examining biological brains, including those of non-human animals, remains a productive source of inspiration and validation for AI [3]. Our use of Q-learning to model a behavior originally observed in cuttlefish sits comfortably within this tradition.

A specific component of our model, the patience function, is adapted from [15]. Their model was developed for robot and self-driving vehicle navigation, where an agent must decide how long to wait to find a waypoint before re-planning to another. We retain the sigmoid form of their patience term but simplify it for the present task; the mathematical details are given in Sect. 3.

2.3 Neurorobotics and Cuttlefish-Inspired Models

Neurorobotics is concerned with embodying brain-inspired controllers in physical robots so that the resulting agent's behavior arises from the joint operation of neural model, body, and environment [4,6]. Krichmar and Hwu [7] articulated a set of design principles for this approach, organized around the requirements that organisms must react quickly to events, must learn and remember over their lifetime, and must weigh options when survival is at stake. Delayed gratification behavioral demonstrations naturally fit within these requirements, which is one reason it is a useful target for a neurorobotic study.

The present work is part of an ongoing program of cuttlefish-inspired research in our group. Pfeiffer et al. developed CuttleBot, a physical robot whose mechanical design and skin-like display are modeled on cuttlefish morphology and signaling [10]. The CuttleBot focuses on the body and its expressive surface; it does not address cuttlefish cognitive behavior. Closer in spirit to the present paper, Kandimalla et al. modeled cuttlefish episodic-like memory in simulation, using a compact architecture inspired by hippocampal indexing theory [5]. The present paper extends that line of work in two directions: it targets a different cognitive capacity, namely delayed gratification rather than episodic memory, and it moves from pure simulation to a physical robot. More broadly, our group has also studied compact memory architectures for online and continual learning [1], reflecting a shared interest in how minimal, biologically motivated mechanisms can support rich behavior.

3 Methods

3.1 Task Design

The task was inspired by the cuttlefish delayed-gratification protocol of Schnell et al. [11]. On each trial the agent was in one of four discrete states corresponding to a left/right arrangement of two prey items. Two states were experimental, in which one option was a delayed high-value live shrimp and the other was an immediately available low-value dead shrimp. The remaining two states were control, in which one option was unobtainable and the other was an immediately available dead shrimp. Side assignments were randomized across trials so that the agent learned reward values associated with prey rather than a fixed side preference. Reward magnitudes are 5.0 for the live shrimp, 1.0 for the dead shrimp, and 0.5 for the unobtainable option. Table 1 summarizes the four task states and the optimal action for each.

Table 1. Task states used in the delayed-gratification experiment.

State	Condition	Left option	Right option	Best action
State 0	Experimental	Live shrimp	Dead shrimp	Left
State 1	Experimental	Dead shrimp	Live shrimp	Right
State 2	Control	Unobtainable	Dead shrimp	Right
State 3	Control	Dead shrimp	Unobtainable	Left

3.2 Q-Learning Agent

The agent learned through Q-learning. It maintained a Q-table of size 4 (states) $\times$ 2 (actions: LEFT, RIGHT), where each entry $Q(s, a)$ stores the expected

reward for taking action a in state s. After every trial, the entry was updated using the prediction error between the reward received and the value expected:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\,[\,r - Q(s,a)\,] \tag{1}$$

where the learning rate is $\alpha = 0.10$, s is the state, a is the action, and r is the reward. The reward was 5.0 for obtainable live shrimp, 1.0 for dead shrimp, and 0.5 for unobtainable live shrimp. The entries converged over many trials to stable values reflecting the underlying reward of each option. Actions were selected via a softmax distribution over $Q(s,\cdot)$ with a temperature $\beta = 1.0$.

3.3 Patience Model

To model time-dependent willingness to wait, we applied a delay-dependent discount only to the live-shrimp Q-value in experimental states. The form was adapted from serotonin experiments in the mouse [9], and a patience model for robot navigation [15]. It was simplified by removing the neuromodulatory gain term and re-tuned for the present task. Letting t denote the delay in seconds, the probability of waiting is:

$$p_{\text{wait}}(t) = \frac{1}{\exp(\beta_w \cdot \Phi(t;\mu,\sigma))} \tag{2}$$

where $\Phi(\cdot;\mu,\sigma)$ is the cumulative normal distribution function, $\mu = 70$ seconds, $\sigma = 20$ seconds, and $\beta_w = 2$. This function is close to 1 for short delays (the agent is willing to wait), drops off through the inflection at $t = \mu$, and decays smoothly toward 0 for long delays. On each experimental trial, the live-shrimp Q-value is multiplied by $p_{\text{wait}}(t)$ before the softmax action selection. Crucially, the discount is applied only to live-shrimp choices, so dead-shrimp and unobtainable Q-values are unaffected. This isolates delay-discounting from baseline learning of reward identity.

3.4 Computational Training and Testing

The agent was first trained for 100 trials with p_{wait} fixed to 1.0, allowing it to learn that live shrimp were preferable to dead shrimp, which were preferable to unobtainable shrimp. The agent was then tested across 13 delay values from 10 to 130 s in 10-second increments, with 100 trials at each delay. The agent or robot would either wait until the live shrimp was obtainable before approaching, or quit waiting and immediately approach the dead shrimp. Performance is measured separately for experimental and control trials as the percentage of trials in which the agent selected the better option (live shrimp in experimental trials, dead shrimp in control trials).

3.5 TurtleBot4 Physical Implementation

The TurtleBot4 implementation was used in two ways. First, the Q-table learned in computational simulation was loaded onto a TurtleBot4 Lite mobile robot

using a Raspberry Pi 4B as the central computing unit to test whether a simulation-trained policy could guide embodied chamber selection. Second, the robot was also run with online Q-learning updates during physical trials, producing a separate robot-trained Q-table. This allowed us to compare the policy structure learned in computational simulation with the policy structure obtained during embodied robot training. Navigation was implemented in ROS2 on Ubuntu, using the platform's built-in Simultaneous Localization and Mapping (SLAM) stack for mapping and localization. The robot used visual cues to identify reward type and chamber availability during the physical task. Prey items were represented by colored markers detected through the TurtleBot4's onboard camera: yellow indicated the live shrimp, corresponding to the high-value delayed reward, and red indicated the dead shrimp, corresponding to the low-value immediate reward. A purple marker placed on the yellow chamber's sliding door allowed the robot to visually detect whether the door was open or closed, signaling the availability of the delayed reward. Together, these visual cues allowed the robot to perceive the task state from its environment and approach the chamber corresponding to the action selected from its Q-table.

It should be noted that the physical embodiment implemented a reduced version of the simulated task. Whereas the computational simulation included four task states, including two experimental states and two control states, the robot implementation focused on the two experimental states in which the live and dead shrimp rewards appeared on opposite sides. The control states involving an unobtainable reward were excluded from the physical trials because they would have required additional visual cue combinations to represent reward type, chamber availability, and obtainability. Given the limitations of the current camera-based recognition pipeline, the physical task was constrained to the two states that could be detected most reliably. This allowed the robot to demonstrate vision-guided delayed-reward choice behavior while maintaining stable perception and navigation performance.

3.6 Test Environments

Initial computational simulations were run using the Python programming environment, as described in Sect. 3.4. We then further validated the full perception–decision–action loop in a Gazebo Harmonic simulation running on ROS2 Jazzy before deploying on the physical robot (see https://youtu.be/UuVfRCssCz0).

Arena and Navigation. The simulated arena (Fig. 1) replicates the physical environment: a central starting area, a narrow gap, and two chambers. A yellow marker (live shrimp) is placed at the left chamber at $(-0.8,\ 2.0)$ m and a red marker (dead shrimp) at the right chamber at $(0.8,\ 2.0)$ m. Four spatial zones—CENTER, GAP, LEFT_CHAMBER, and RIGHT_CHAMBER—are defined by position and used to confirm the robot reached its chosen chamber before updating the Q-table. In each simulated trial, the red and yellow markers were placed according to the sampled task state, chosen randomly.



Fig. 1. Left: top-down view of the simulated arena in Gazebo. Right: RViz2 visualization showing the occupancy map, AMCL particle cloud, and MPPI trajectory during navigation.

Software Architecture. Four custom ROS2 nodes communicate via JSON-encoded messages. The *State Manager* orchestrates the trial loop, commanding navigation and determining the task state from vision. The *Brain Node* wraps the Q-learning agent, receiving the current p_{wait} and vision-determined state and returning a softmax-selected action. The *Vision Node* processes frames from a simulated OAK-D camera (320×240), detecting red and yellow regions via HSV thresholding. The *Location Awareness* node classifies the robot's pose into spatial zones.

State Determination and Trial Flow. As the robot navigated toward the gap, it passively observed color markers through the onboard camera. When both colors were visible in the same frame, the horizontal positions of their centroids determine the left–right arrangement, which was combined with a random assignment to select one of the experimental task states (Table 1). Each trial then proceeds: the agent selected an action, the robot navigated to the chosen chamber, and the Q-table was updated after food was chosen.

Physical Testing Environment. The physical testing environment was designed to approximate a shallow marine habitat while supporting the delayed gratification task (Fig. 2). The arena consisted of two custom wooden chambers with transparent acrylic sliding doors, allowing the prey markers to remain visible from outside the chamber. Additional fabricated seagrass, rocks, and coral reef structures were placed throughout the arena to provide visual complexity and marine-inspired context.

4 Results

4.1 Training

After 100 training trials in the computational simulation with $p_{\text{wait}} = 1.0$, the agent learned the correct action direction in all four states, but the strength of the



Fig. 2. Artificial reef environment with shrimp chambers. The Turtlebot is waiting for the door of the yellow chamber to open. (Color figure online)

learned policy differed across conditions (Fig. 3). Table 2 shows the final learned policy. In every state, the action with the higher action selection probability matched the appropriate action listed in Table 1.

Table 2. Final learned policy after 100 training trials in computational simulation. Q-values were converted into action selection probabilities using $\beta = 1.0$ in the Softmax function. Bold percentages indicate the higher-probability action.

State	Condition	Q(LEFT)	Q(RIGHT)	P(LEFT)	P(RIGHT)	Preferred action
State 0	Experimental	4.71	0.27	**98.8%**	1.2%	Left/live shrimp
State 1	Experimental	0.57	3.73	4.1%	**95.9%**	Right/live shrimp
State 2	Control	0.40	0.65	43.8%	**56.2%**	Right/dead shrimp
State 3	Control	0.75	0.34	**60.1%**	39.9%	Left/dead shrimp

In the experimental states, the learned policy strongly favored the live-shrimp option regardless of side, and in the control states the learned policy had a slight preference toward the obtainable dead shrimp. In State 0, where the live shrimp appeared on the left, the probability of selecting LEFT was 98.8%. In State 1, where the live shrimp appeared on the right, the probability of selecting RIGHT was 95.9%. This indicates that the agent learned reward identity rather than a fixed spatial preference. The control states favored the correct obtainable dead-shrimp option, but with weaker action separation. This weaker preference is consistent with the reward structure of the task. The prey choice could have been made more decisive with parameter tuning of either the Softmax or the reward values. However, the values used in the present experiments had a good mix of exploration and exploitation. Across 30 independent computational simulation runs, the same pattern appeared in the training curves

(Fig. 3). Experimental-state performance increased steadily over training, while control-state performance remained closer to chance but directionally favored the obtainable dead shrimp. Together, these results show that the agent learned the task structure while producing stronger policies when reward contrast was larger.

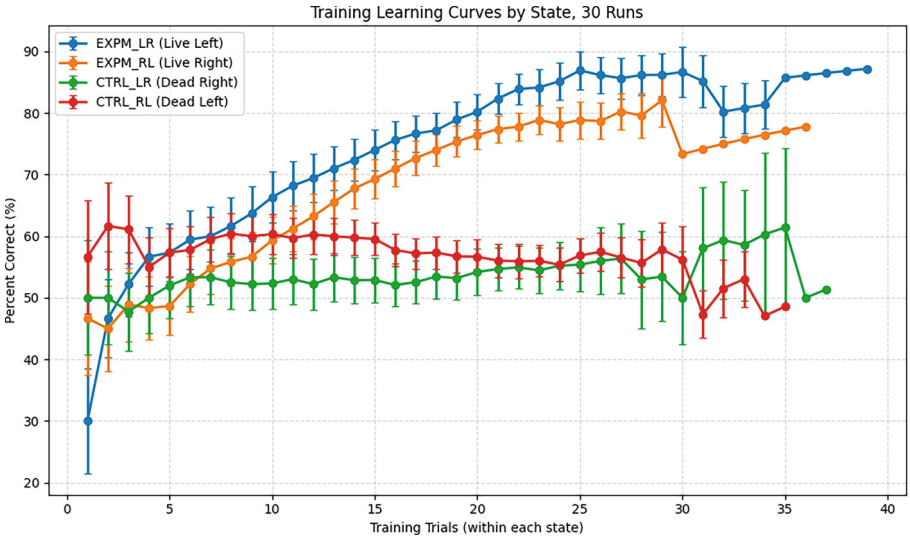


Fig. 3. Training performance across 30 computational simulation runs. Each curve shows the mean percentage of trials in which the agent selected the optimal action for a given task state during training. Error bars indicate standard deviation across runs. Experimental states required choosing the live shrimp, whereas control states required choosing the obtainable dead shrimp.

4.2 Delayed Gratification Across Delay Seconds

The delays and responses shown in Fig. 4 were similar to those reported in the cuttlefish (see Fig. 2 of [11]) in which the agent or cuttlefish would wait for a live shrimp to become available up to around 60 s. In experimental trials, the agent selected the delayed live shrimp at a high rate for short delays, but this preference declined as the delay increased. The steepest decline occurred near the midpoint of the patience function, indicating that delay-dependent discounting specifically reduced the effective value of the live shrimp. As with the cuttlefish experiments, the delay did not effect the choice in the control condition. Control performance remained relatively stable across delays because the patience function was not applied to the dead-shrimp or unobtainable options. Thus, increasing delay selectively reduced preference for the delayed high-value reward rather than producing a general decline in task performance.

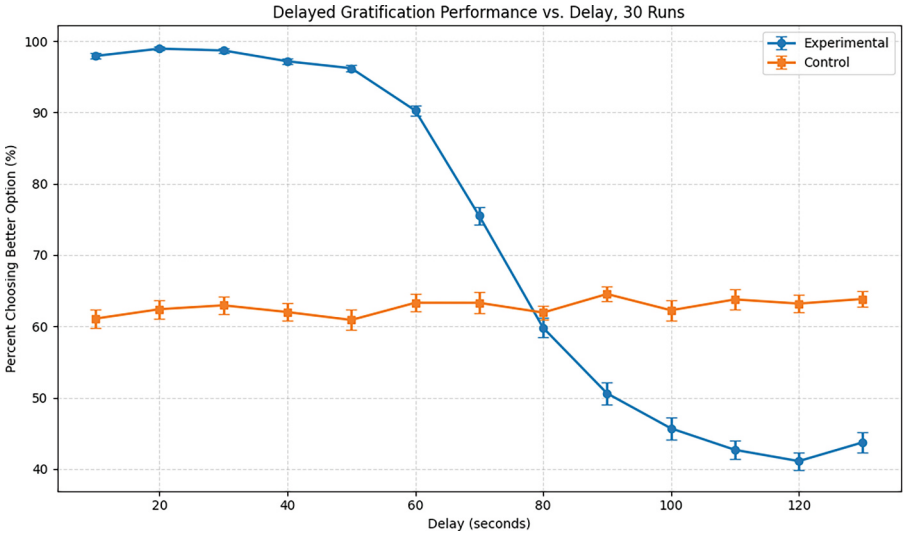


Fig. 4. Delayed-gratification performance across delay durations. Curves show the mean percentage of trials in which the agent selected the better option across 30 computational simulation runs, with error bars indicating standard deviation. In experimental trials, the better option was the delayed live shrimp. In control trials, the better option was the immediately available dead shrimp.

4.3 Robotic Transfer and On-Robot Training

The TurtleBot4 experiments evaluated both simulation-to-robot policy transfer and on-robot learning. In the transfer condition, the Q-table learned during computational simulation was loaded onto the robot, allowing the TurtleBot4 to navigate toward the chamber selected by the simulation-trained policy. In the on-robot training condition, the robot ran the task with online Q-learning updates, producing a separate robot-trained Q-table.

In the physical robot trials, chamber locations were specified using predefined coordinates, allowing the robot to navigate toward the selected chamber. Color recognition was used for the experimental states, in which yellow and red markers indicated the live and dead shrimp options, respectively. Because the control states required representing unobtainable prey in addition to prey identity, the full visual recognition pipeline was limited to the two experimental states.

The robot-trained Q-table showed the same qualitative structure as the simulation-trained policy (Table 3). In both computational simulation and on-robot training, the experimental states produced strong correct-action probabilities, whereas the control states produced weaker but directionally correct preferences. This pattern suggests that on-robot learning reproduced the main policy structure observed in simulation. However, because the control-state robot trials used predefined chamber coordinates rather than full color-based recogni-

tion of unobtainability, these results should be interpreted as coordinate-guided embodied learning rather than full visual recognition of all four task states.

Table 3. Comparison of correct-action softmax probabilities from computational simulation training and on-robot training. The robot-trained Q-table was generated during physical trials with online Q-learning updates.

State	Correct action	Simulation	Robot training
State 0	Left/live shrimp	98.8%	95.8%
State 1	Right/live shrimp	95.9%	97.5%
State 2	Right/dead shrimp	56.2%	58.4%
State 3	Left/dead shrimp	60.1%	61.9%

5 Conclusions

Cuttlefish and higher order vertebrates demonstrate self-control in the form of delayed gratification. This is one of the hallmarks of cognition [2,12]. To investigate this ability, we built a neurorobotic system that demonstrates delayed gratification. A Q-learning agent learned reward values in a four-state foraging task inspired by cuttlefish [11]. A simulation-trained policy was transferred to a TurtleBot4, and separate on-robot training produced a qualitatively similar learned policy. The present results show that a simple Q-learning agent, paired with a delay-dependent patience function, can reproduce the self-control abilities observed in cuttlefish. They also show that this behavior can run on a real robot using standard navigation and vision tools, in line with the broader neurorobotics goal of grounding biological models in physical hardware.

There are still limitations. The patience function describes waiting behavior but does not model how the brain actually tracks time; adding a biologically motivated timing mechanism is a natural next step.

Code Availability. The source code for the simulations and TurtleBot4 implementation is available at: https://github.com/v2pir/turtlebot.

Acknowledgments. This was supported by the UC Irvine California Institute for Telecommunications and Information Technology (CALIT2) in collaboration with the UC Irvine Undergraduate Research Opportunities Program (UROP). J.L.K. was supported in part by National Institute of Neurological Disorders and Stroke award R01NS135850-02.

References

1. Alonso, N., Krichmar, J.L.: A sparse quantized hopfield network for online-continual memory. Nat. Commun. **15**(1), 3722 (2024). https://doi.org/10.1038/s41467-024-46976-4
2. Birch, J., Schnell, A.K., Clayton, N.S.: Dimensions of animal consciousness. Trends Cogn. Sci. **24**(10), 789–801 (2020). https://doi.org/10.1016/j.tics.2020.07.007
3. Hassabis, D., Kumaran, D., Summerfield, C., Botvinick, M.: Neuroscience-inspired artificial intelligence. Neuron **95**(2), 245–258 (2017). https://doi.org/10.1016/j.neuron.2017.06.011
4. Hwu, T., Krichmar, J.: Neurorobotics: connecting the Brain. Body and Environment. MIT Press, Cambridge, MA (2022)
5. Kandimalla, S., Wong, Q.Y., Zheng, K., Krichmar, J.L.: Episodic-like memory in a simulation of cuttlefish behavior. Sci. Rep. **16**(1), 2169 (2025). https://doi.org/10.1038/s41598-025-31950-x
6. Krichmar, J.L.: Neurorobotics–A thriving community and a promising pathway toward intelligent cognitive robots. Front. Neurorobot. **12**, 42 (2018). https://doi.org/10.3389/fnbot.2018.00042
7. Krichmar, J.L., Hwu, T.J.: Design principles for neurorobotics. Front. Neurorobot. **16**, 882518 (2022). https://doi.org/10.3389/fnbot.2022.882518
8. Mischel, W., Ebbesen, E.B.: Attention in delay of gratification. J. Pers. Soc. Psychol. **16**(2), 329–337 (1970). https://doi.org/10.1037/h0029815
9. Miyazaki, K., Miyazaki, K.W., Yamanaka, A., Tokuda, T., Tanaka, K.F., Doya, K.: Reward probability and timing uncertainty alter the effect of dorsal raphe serotonin neurons on patience. Nat. Commun. **9**(1), 2048 (2018). https://doi.org/10.1038/s41467-018-04496-y
10. Pfeiffer, M.A., et al.: Cuttlebot: emulating cuttlefish behavior and intelligence in a novel robot design. In: Brock, O., Krichmar, J. (eds.) From Animals to Animats 17. pp. 93–105. Springer Nature Switzerland (2025). https://doi.org/10.1007/978-3-031-71533-4_7
11. Schnell, A.K., Boeckle, M., Rivera, M., Clayton, N.S., Hanlon, R.T.: Cuttlefish exert self-control in a delay of gratification task. Proceedings of the Royal Society B: Biological Sciences **288**(1946), 20203161 (2021). https://doi.org/10.1098/rspb.2020.3161
12. Schnell, A.K., Clayton, N.S.: Cephalopods: ambassadors for rethinking cognition. Biochem. Biophys. Res. Commun. **564**, 27–36 (2021). https://doi.org/10.1016/j.bbrc.2020.12.062
13. Schultz, W., Dayan, P., Montague, P.R.: A neural substrate of prediction and reward. Science **275**(5306), 1593–1599 (1997). https://doi.org/10.1126/science.275.5306.1593
14. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA, second edn (2018)
15. Xing, J., Zou, X., Krichmar, J.: Neuromodulated patience for robot and self-driving vehicle navigation. In: arXiv preprint arXiv:1909.06533 (2019). https://doi.org/10.48550/arXiv.1909.06533