Cognitive Computation and Systems

Research Article

Advantage of prediction and mental imagery for goal-directed behaviour in agents and robots

eISSN 2517-7567
Received on 9th September 2018
Revised 30th November 2018
Accepted on 20th December 2018
doi: 10.1049/ccs.2018.0002
www.ietdl.org

Jeffrey L. Krichmar[1,2] ✉, *Tiffany Hwu*[1], *Xinyun Zou*[2], *Todd Hylton*[3]

[1]*Department of Cognitive Sciences, University of California, Irvine, USA*
[2]*Department of Computer Science, University of California, Irvine, USA*
[3]*Department of Electrical and Computer Engineering, University of California, San Diego, USA*
✉ *E-mail: jkrichma@uci.edu*

Abstract: Mental imagery and planning are important aspects of cognitive behaviour. Being able to predict outcomes through mental simulation can increase environmental fitness and reduce uncertainty. Such predictions reduce surprise and fit with thermodynamically driven theories of brain function by attempting to reduce entropy. In the present work, the authors tested these ideas in a predator–prey scenario where agents with a limited energy budget had to maximise food intake, while avoiding a predator. Forward planning agents, with the ability to mentalise, to Actor Critic agents that do not plan beyond the current state were also compared. The authors show that the ability to mentalise has distinct advantages when in noisy, uncertain stimuli. These advantages are even more prevalent when tested in the real world on physical robots. The authors' results highlight the importance of taking into consideration mental imagery and embodiment when constructing artificial cognitive systems.

1 Introduction

It has been suggested that any self-organising system that is at equilibrium with its environment must minimise its free energy [1]. In other words, the system must adapt or evolve to resist a natural tendency towards disorder in an ever-changing environment [2]. Organisms must minimise the long-term average of surprise, which is the inverse of entropy, by predicting future outcomes, so that they minimise the expenditures required to deal with unanticipated events. The idea of minimising free energy has close ties to many existing brain theories, such as Bayesian brain, predictive coding, cell assemblies and Infomax, as well as an evolutionary inspired theory called Neural Darwinism or neuronal group selection [1]. The notion of surprise, in our case, is the unexpected uncertainty of an event [3].

In the theory of neuronal group selection [4], plasticity is modulated by value, which is signaled by neuromodulatory systems such as acetylcholine, dopamine, norepinephrine, and serotonin [5], and hormonal systems [6, 7]. Value systems control which neuronal groups are selected and which actions lead to evolutionary fitness, that is, predicting outcomes that lead to positive value and avoid negative value. In this sense, predicting value is inversely proportional to surprise. From a non-equilibrium thermodynamic perspective, value is associated with those actions that minimise the increase of entropy (e.g. feeding, predicting outcomes, gathering information). Such actions include energy-efficient movement, energy-efficient alert sensory scanning of the environment, and foresight associated with organisms having higher cognitive functions.

Inspired by evolution of biological organisms on multiple timescales, we propose an architecture that is a closed-loop system in which the control (algorithm) is closely coupled with the body (robots) and the world (environment). The schematic in Fig. 1 shows the overall architecture for an efficient bio-inspired agent. The agent has innate values, which can be positive (e.g. food, rewards, progress) or negative (e.g. energy expenditure, pain, frustration). These values are derived from the environment and value systems, such as neuromodulation and hormones, which send signals to the brain to adapt behaviour. Since the expected value is thought to be inversely proportional to surprise [1], predicting value is key to the agent's fitness. Since the world is dynamic, the agent must adapt its behaviour to survive. Fitness evaluation in Fig. 1 is the metric for evolving these algorithms, which can be how long the system can perform without intervention, how successful the system is in the task environment, and how energy efficient the system is in performing a task. These agents must adapt their computation and resource allocations to survive.

A number of researchers, especially in robotics [8–12], as well as in neuroscience [13], have stressed the importance of embodiment when testing models of cognitive behaviour. In some cases, they have stressed that intelligent behaviour can be observed without representation or reasoning [14, 15]. Indeed, behaviour-based robots following these ideas have shown impressive performance. However, the brain does contain what many call internal models to predict outcomes and plan [16]. While these are not necessarily symbolic representations, as in classical artificial intelligence systems, they are still neural representations. Brain areas such as the frontoparietal cortex [17], motor cortex [18], insula cortex [19, 20], are necessary for prediction, planning, and awareness. Therefore, it is important to examine the effects of planning in an embodied model.

To demonstrate this idea, we designed a predator–prey scenario, where the agent of interest wants to maximise positive value (i.e. acquiring food), while minimising negative value (i.e. being eaten by a predator or starving). The agent was further constrained by having a finite energy budget. If the agent wandered around too long without obtaining food, it starved.

By recursively iterating through possible outcomes and their value, the predictive engine in our agent has the ability to perform mental imagery. These predictions take into account the actions the predator may take in response to an agent action. Mental imagery has been shown to influence future actions, and is an important aspect of theory of mind; the ability to understand and predict the intentions of others [21]. This awareness of self and others would be critical component for a conscious organism. It is also important for the development of artificial cognitive systems [22, 23].

In the present work, we will test our planning, value-driven model against other candidate agents in simulation. Furthermore, we will test the algorithm in an autonomous robot to show the advantage of mental simulation in dynamic, real-world situations, and to test whether different behavioural strategies emerge when the agent is embodied. The hypothesis is that such predictions can lead to better performance, especially when information is unreliable as in noisy physical environments.

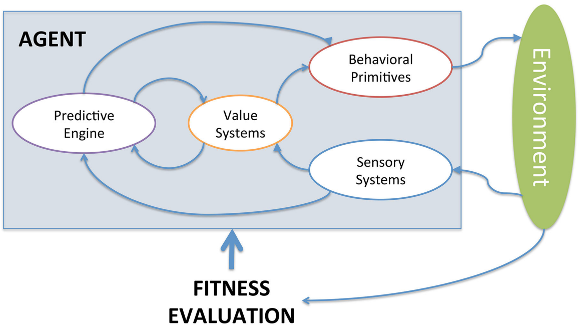

Fig. 1 *Architecture for robots and algorithms. The agent must take actions to maximise fitness in a complex, dynamic environment. It is endowed with innate values and behavioural primitives. Based on its sensory input, it must evolve to predict outcomes that maximise positive value and minimise negative value. The value system can influence predictions by setting context or dynamically adjusting values*

2 Methods

We designed a predator–prey scenario for simulation and robot experiments. In the scenario, there is a behaving agent, a predator that chases the agent, a nest area where the agent is safe from the predator, and a food source that is outside the nest area. The agent has three behavioural primitives: (i) find food, (ii) find nest, and (3) avoid a predator. The agent must choose between these three actions to acquire energy and avoid being eaten. The agent also has a limited energy budget that decreases with time. This decrease is faster during exploration than when in the nest. We will first describe the specifics of the scenario for simulations, and then describe how this was carried out in the robot experiments.

2.1 Simulation experiments

Simulations were conducted on a 10 × 10 grid environment. The nest was always in the same corner of the environment, that is, Cartesian coordinates (1,1). The positions of the agent and the predator were chosen randomly on the first trial. A trial consisted of the predator and agent making moves until the agent found food, was caught, or starved. If the agent found food, its energy level was replenished and the next trial began with the agent and predator starting from their last positions on the previous trial. If the agent was caught by the predator or starved, the positions of the agent and the predator were chosen randomly for the next trial. The food location changed on each trial. On a given time step, the agent would: (i) Seek food by moving closer to the food source, (ii) Seek its nest by moving closer to the nest location, (iii) Avoid the predator by moving away from the predator agent. A move, which could be in any of eight directions, moved the agent one grid location closer to its desired position. To make sure that as many configurations as possible were tested, 500 trials were conducted for 10 random seeds (5000 total trials).

We tested three types of agents: (i) Random – the agent chose randomly between the three actions. (ii) Actor Critic – an actor critic model that learned the value of state-action pairs, and (iii) Planning – a Q-learning-based planning agent that attempted to predict the value of outcomes by recursively iterating over state-action pairs. The distances to the food source, nest, and predator define the state space. Since the longest distance is from one corner of the grid to the other, the state space is the Euclidean distance from (1,1) to (10,10), which equals 12.73, rounded up to 13. Therefore, the total state space is: 13 × 13 × 13. Although this state space does not capture directionality or occlusions in simulation, the agents will need to take this into account in the robot experiments. We ensured that each type of agent encountered the same configurations by resetting the random number sequence and storing the 500 game configurations before trials began.

The predator followed a fixed behaviour pattern. In the case of the simulation, the predator moved at half the speed of the agent. The predator would move towards the agent if the agent was less than a Euclidean distance of 6 from the agent (i.e. it could sense the agent) and >4 from the nest. The predator moved away from the nest if it was ≤4 from the nest. Otherwise, the predator moved in a random direction.

At the start of a trial, the agent's energy level started at 5. If the energy level dropped below 0.001, the agent starved. After each time step, the agent's energy level decreased:

$$\begin{aligned} E(t+1) &= E(t) - 0.05; \quad \text{if agent in nest, distance} < 4.0 \\ E(t+1) &= E(t) - E(t)0.10; \quad \text{otherwise} \end{aligned} \tag{1}$$

This had the effect of a slower constant drop in energy when in the nest and a more rapid drop in energy when foraging. The assumption being that resting in the nest takes less energy than exploring and avoiding predators in the environment.

2.1.1 Random agent: At every time step in the simulation, the random agent chose from one of the three actions (find food, find nest, avoid predator). The action was carried out and then a new action was chosen randomly at the next time step.

2.1.2 Actor-critic agent: For the Actor Critic agent, we implemented a temporal difference learning actor-critic model [24]. The Actor Critic agent was model free and selected actions based on its past experience. It did not take into account the actions or the state of the predator agent. The actor critic only used the current state to dictate an action. Similar to other critic models, the critic learned expected values for different states, and the actor learned appropriate mappings for actions given a particular state. The critic and actor were updated after every move. The delta rule was based on a reward prediction

$$\delta(t) = R(t) + C(s,t) - C(s,t-1); \tag{2}$$

where s is the state at time t, $R = -5$ if caught by the predator, −5 is starved, $(5.0 - E(t))$ if food is found, and 0 otherwise. C was initially set to near zero (i.e. 2.22–16).

The critic was as follows:

$$C(s,t+1) = C(s,t) + 0.25\delta(t); \tag{3}$$

The actor was updated as follows:

$$Q(s,a,t+1) = Q(s,a,t) + 0.25\delta(t); \tag{4}$$

where Q is the actor, and a is the action to be taken at time $t+1$. Each test case had 500 trials with a number of time steps within a trial. This was thought to be sufficient for the state space to be learned. To verify, we tested a range of learning rates and number of trials and found that the learning rate of 0.25 and 500 trials was sufficient for learning. Values <0.25 resulted in a too little learning and values >0.25 resulted in overlearning, which led to erroneous, perseverative action selection. Q was initially set to near zero (i.e. 2.22–16).

A Softmax function with a temperature, β, equal to 5 was used to select an action based on the expected values Q. A range of temperatures from 0.1 (exploration) to 10.0 (exploitation) was tested. The overall performance was similar between 2 and 10

$$p_i = \frac{e^{-\beta Q(s,a_i)}}{\sum_{j=1}^{3} e^{-\beta Q(s,a_j)}}; \tag{5}$$

where p_i is the probability of taking action i out of the three possible actions and β is the temperature.

2.1.3 Planning agent: The planning agent simulated mental imagery by iterating through the cause and effect of its actions, taking into account what it predicted the predator would do in response to its actions. The Planning agent searched a tree of potential actions by itself and the predator to predict the best course of action. This was a recursive algorithm, but since the state space was small, all possible outcomes were predicted in a tractable number of steps. Unlike the Actor Critic agent, the Planning agent

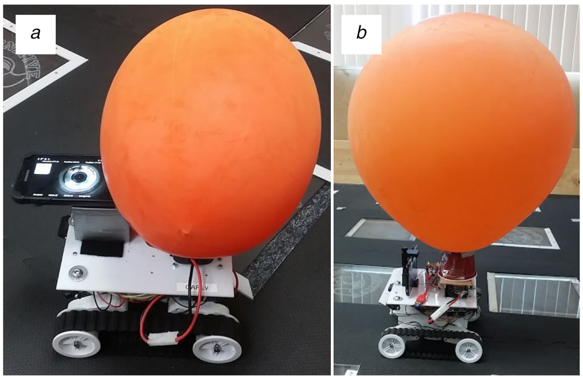

Fig. 2 *Robots used in the experiments. Both robots had a Samsung Galaxy 5 smartphone, an IOIO interface board, and 4 IR sensors mounted on a Plexiglas base. The Plexiglas was attached to a Rover 5 robot. The smartphones could send motor commands and read IR sensors via a Bluetooth connection with the IOIO. The orange balloons on the robots allowed them to be seen in the arena*

(A) Agent. The agent had a panoramic mirror attached to its smartphone camera. The camera was facing downward to provide a 290° view of the environment, ***(B)*** Predator. The predator used the camera on a forward facing smartphone to locate objects

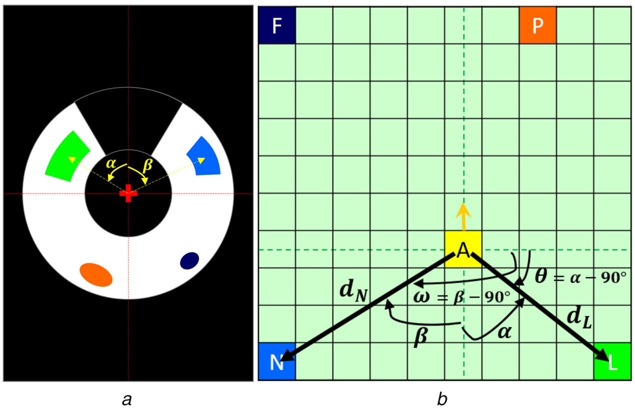

Fig. 3 *Localisation of agent and object positions in robot arena*

(A) Camera view of the smartphone after attaching the panoramic camera. The agent had a 290° view of the environment. It used color blob detection to locate markers in the arena, the predator, and the food source, ***(B)*** Once objects were located, the agent (A) could estimate its own position and the distance to food (F), nest (N), and predator (P). The arrows show the agent's heading and it's bearing to the corners of the environment

does not learn the state space. Rather, it makes estimates of possible outcomes. An expected value Q is calculated for each action

$$Q(s, a, t + 1) = \prod_{s'} (T(s, a, s')^{*}[R(s') + 0.99^{*} \max (Q(s', a')]); \tag{6}$$

where $T(s,a,s')$ is the probability of transitioning from state s to state s' for action a, R is the same as in the Actor Critic agent, and $Q(s',a)$ is the max expected value at state s'. The equation iterates recursively until an end condition is met (i.e. food, predator, starve). For each iteration, both the agent and the predator simulate an action. The transition probability, T, is derived from the corresponding Q values for each action using the Softmax function given in (5). Equation (6) is similar to the Forward model in [25], however, rather than the sum we used the product, discounted to get the overall expected values of different actions from a given state. After the expected value for each action from the current state is calculated, an action is selected using the Softmax function given in (5).

2.1.4 Testing under certain and uncertain conditions: Both the Actor Critic and the Planning agents were tested under certain and uncertain conditions. In the certain conditions, the positions of the food and predator (i.e. the current state) were known. The uncertain condition simulated unreliable sensory information. In the uncertain condition, the position of an object (i.e. food or predator) was known if it was within a distance of 3, in which case the mean of the object was the object's current position, and the standard deviation, s, was set to 0.25. Otherwise, the mean was set to the last known position and s increased by 0.25, which in effect, increased the uncertainty of the object's position. The standard deviation was not allowed to be > 13. Every time step, the positions of the food source and the predator were drawn from a normal distribution with the mean and standard deviation as described above. It was assumed that the agent knew its position and the position of the nest.

2.2 Robot experiments

Even though we introduced uncertainty in the simulations, the noise does not reflect the unreliability of real sensors in physical environments. Moreover, the simulation experiments had a ‘God's eye’ view of the world, and agents made omnidirectional movements without consideration of the cost and time of movements.

Therefore, we ran the model with autonomous mobile robots, where the prey robot ran either the planning or random agent model. It would have taken too long for the Actor Critic to learn the state space in a robot experiment. The predator robot followed the same fixed strategy as in the simulations. The robot experiments were conducted in an indoor arena that was 3 m by 3 m. Camera vision was used to recognise landmarks, agents, and localisation in the arena. The robot experiments were important to test the model with local sensing and movement constrained by physics.

Objects of interest were colour coded. The predator and agent had orange balloons attached to the robots (see Fig. 2), and the food source was a purple balloon. A colored marker in the corner identified the nest area. Another colored marker was used to estimate position in the room (Fig. 3). For both the predator and the agent, we used our Android-based Robots [26]. These robots are constructed from an Android phone, an input output interface board called IOIO, which is connected to the smartphone via Bluetooth or USB, an off-the-shelf robot base (Rover 5 Robot Platform with Tank Treads), and infrared sensors (Sharp Microelectronics IR Sensor) for distance sensing. The robot takes advantage of the sensors on the phone (e.g. camera, compass), as well as additional sensors external to the phone (e.g. IR sensors) via the IOIO. The Android phone interacts with the base's motor controller via the IOIO. The cameras on the robot's smartphones were used for object recognition and IR sensors for collision detection and obstacle avoidance (see Fig. 2). The software application, which computed the algorithm and controlled the robot, was written in Java using Android Studio and deployed on Samsung Galaxy S5 smartphones (The Android application software for the agent robot can be found at: https://gitlab.com/fitany/ABR_FoodPredator, and the software for the predator robot can be found at: https://github.com/UCI-ABR/ABR_predator). OpenCV libraries for Android were used to handle image processing.

The prey agent had a panoramic mirror attached to the smartphone camera to give it a 290° view of the environment (see Figs. 2*A* and 3*A*). Although this gave the agent a wide-angle view, the image resolution was poor due to warping. The distance to objects was estimated as an inverse power function of the size of the minimum enclosing circles for the food, nest, and predator blobs in its field of view. The mirror limited the resolution and accuracy of these estimates. The nest was marked with a blue piece of construction paper, and a corner of the arena was marked with a green piece of construction paper (see Fig. 3). These were used to triangulate the agent's position and its distance to other objects. From the nest's and corner's absolute positions and their distances and angles relative to the agent, the agent could estimate its current location in one of the 10 × 10 grid cells (see Fig. 3*B*). If an object was occluded, the last known distance to the object was used for

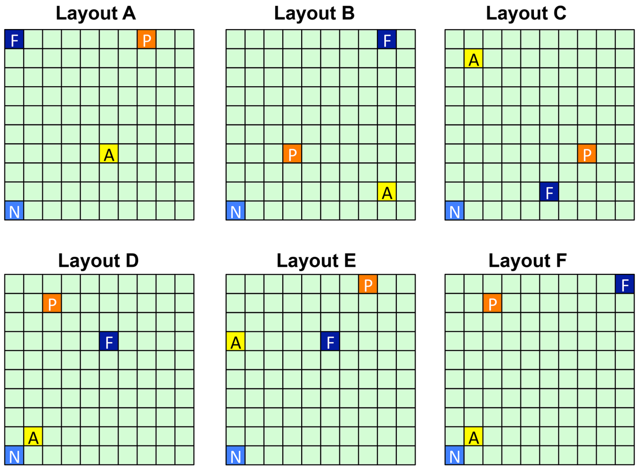

Fig. 4 *Layouts used for robot experiments. The letters denote the initial positions of the nest ('N'), agent ('A'), predator ('P'), and food 'F')*

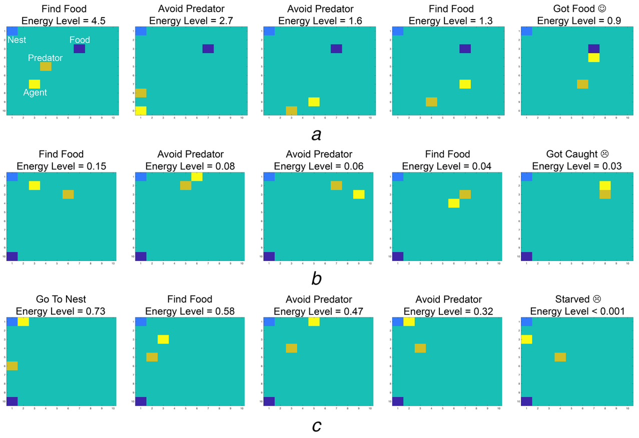

Fig. 5 *Behavioral sequences from the simulations*
(A) Sequence where the gets to the food source, ***(B)*** Sequence where the agent is caught, ***(C)*** Sequence where the agent starves

the state table entry. Each grid corresponded to 30 cm^2. The positions of food and predator were also estimated from their distances and angles relative to the agent and the agent's position. If the agent found food (within 60 cm of estimated distance) or starved, it sent a WiFi message to the predator indicating the trial was over. The random agent algorithm and the planning algorithm, which were described above, guided the agent's behaviour. The state space in the model corresponded to the distance to food, nest and predator in the arena. To allow for the robots to occasionally starve, the initial energy level was reduced to 3. Due to its long training time, the Actor Critic agent was not used in the robot experiments.

The predator robot was an Android-based robot whose actions were governed by a fixed strategy based on its distance to the agent, as well as the agent's distance to the nest (see Table 1). The predator had a forward-facing smartphone and used OpenCV blob detection to locate the prey agent (see Fig. 2*B*). When the prey agent was detected, the predator chased the agent by simultaneously moving forward and turning to center the orange agent blob in its field of view. To match the conditions of the simulation, the predator did not chase the agent when it was far away. That is, when the size of the orange blob detector, which denoted the prey agent, was <8% of the frame width.

The predator and prey robots communicated information between them using the WiFi on the Android smartphones. The prey messages told the predator that it found food, was in the nest, or was away from the nest. The predator messages told the prey when it was caught. Table 1 describes how these messages were handled by the robots.

Additionally, when the agent calculated that it was within 60 cm of the nest, it would send a WiFi signal to the predator to cease chasing behaviour (see Table 1). Another WiFi signal was sent when the agent was >60 cm from home, which caused the predator to resume chasing the agent (see Table 1). When the predator was not chasing the agent, it alternated between turning for a random amount of time and moving forward for a random amount of time. If at any point the size of the orange blob detector was >30% of the camera frame width, the predator sent a signal to the agent indicating that it had been caught, thus ending the simulation (see Table 1).

Table 1 Predator actions

Prey distance to nest or food	Predator blob detection of prey	Prey action	Predator action
$d_{\text{food}} < 60$ cm	N/A	'Found Food' message to predator terminate trial	terminate trial
$d_{\text{nest}} < 60$ cm	N/A	'At Nest' message to predator select action from model	random search
$d_{\text{nest}} >= 60$ cm	blob < 8% of frame width	'Away From Nest' message to predator select action from model	random search
$d_{\text{nest}} >= 60$ cm	blob> = 8% of frame width	'Away From Nest' message to predator select action from model	chase agent
$d_{\text{nest}} >= 60$ cm	blob > 30% of frame width	N/A terminate trial	'Caught' message to agent terminate trial

We used six layouts, which had different starting positions for the food, agent, and predator (see Fig. 4). Both the Planning agent and the Random agent conducted five trials per layout. It should be noted that the algorithms operated in real time and computation was not a factor in slowing down the agent's behaviour in the robot experiments. The robots and food source were placed in their initial positions prior to every trial.

3 Results

3.1 Behaviour in simulation experiments

The simulation environment was dynamic enough for interesting behaviours to emerge between the predator and agent. Fig. 5 shows screenshots from simulation sequences with the Planning agent with uncertainty. In Fig. 5*A*, the agent escapes the predator and gets to the food source. In Fig. 5*B*, the agent is trying to avoid the predator. As its energy level depletes, the agent tries to find the food source. A bad estimate of the food location leads the agent to the predator, which results in the agent being caught. In Fig. 5*C*, the agent cannot get around the predator and towards the food source. Eventually, the agent depletes its energy and starves.

In general, the Planning agent outperformed the other agents by obtaining more food, avoiding the predator and not starving (see Table 2 and Fig. 6). With the exception of the ActorCritic-Uncertain agent compared to the Random agent, and the ActorCritic-Uncertain agent compared to the ActorCritic-Certain agent, all comparisons were highly significant ($p < 0.00001$; two-sample Kolmogorov–Smirnov goodness-of-fit test with Bonferroni correction).

The ability to plan ahead and predict outcomes through mental simulation was advantageous in this environment. Specifically, Planning agent always outperformed the Actor Critic agent. When the positions of the objects were known (Planning-Certain in Fig. 6), the Planning agent was near perfect; that is, it rarely was caught or starved, and was able to plan strategies that got the agent to food sources. The Planning agent under uncertain conditions was significantly better than the Actor Critic agent, even when the Actor Critic agent knew the position of objects (compare Planning-

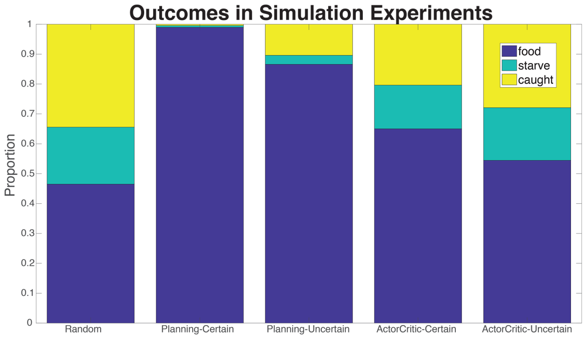

Fig. 6 *Behavioural outcomes in the simulations. For every trial, an outcome for the agent could be found food, starved, or caught by predator. The proportions of these three outcomes are shown in the figure. Each agent ran for 500 trials with 10 random sequences for a total of 5000 trials*

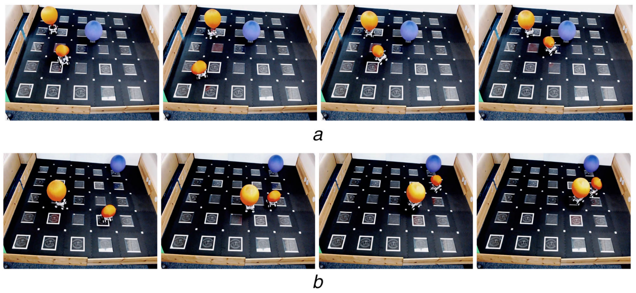

Fig. 7 *Behavioural sequences from the robot experiments. The agent and predator had orange balloons attached to the robot. The food source was a purple balloon*

(A) Sequence where the robot avoided the predator and found food, ***(B)*** Sequence where the predator caught the robot

Uncertain to ActorCritic-Certain). This suggests that the ability to plan ahead outweighs perfect knowledge of sensory surroundings.

Planning ahead and mental simulation were beneficial in uncertain situations. In the simulations, uncertainty is defined as a noisy estimation of object positions in the environment (i.e. food and predator). The Planning agent under these uncertain conditions performed well, finding food on 87% of the trials (Planning-Uncertain in Fig. 6). Interestingly, this agent starved only 3% of the time, meaning that the agent would rather take the chance of being caught to find food, than starve. The Actor Critic agent did not perform as well in these uncertain conditions. In fact, it was not significantly better than the Random agent (compare ActorCritic-Uncertain to Random in Fig. 6). Taken together, these simulations demonstrate that the ability to predict outcomes can be advantageous in dynamic, uncertain environments.

3.2 Behavior in robot experiments

The robot experiments added new dimensions because the agents needed to physically move in a real environment and use noisy sensors to locate objects of interest. Fig. 7 shows screenshots from the robot experiments. In Fig. 7*A*, the agent avoids the predator and goes to the nest. While in the nest, the predator turns away from the agent. The agent takes advantage of this opportunity by heading straight to the food source. In Fig. 7*B*, the agent is avoiding the predator and moving towards the food source. However, as it nears the food source, the agent makes a bad decision, either due to the stochastic action selection rule or a noisy sensor, and the predator is able to catch the agent.

Similar to the simulation results, planning ahead was beneficial in the robot experiments (see Table 3 and Fig. 8). The Planning robot found food on 77% of the trials compared to 33% for the Random robot ($p < 0.005$; two-sample Kolmogorov–Smirnov goodness-of-fit test). Interestingly, the Planning robot never starved, whereas the Random robot starved on 50% of the trials. Since the Random robot was incapable of putting together sequences of goal-driven actions, it tended to wander or stay in one region of the environment too long. Moreover, this suggests the Planning robot was planning paths towards food before its energy reached a critically low level. However, these actions put the robot at risk for being caught by a predator.

The robot experiments also emphasised differences between a virtual simulation and a physical experiment. In the simulations, the agent can instantly move in any direction. For a robot, especially one with a scrub steering mechanism, turning and moving takes time. In the present experiments, this allowed the predator to approach while the agent was carrying out an action. Sensing in a real environment, even in a constrained environment, is noisy. Oftentimes the robots would sense an object of interest on one time step, only to lose it on the next. Interestingly, the food source was large enough to occlude the robots from seeing each other. This not only led to the agent being able to hide behind the food source, but also resulted in the agent not knowing the predator was near. These types of unforeseen situations emerge from running algorithms on embodied agents (Pfeifer *et al.*, 2014).

3.3 Effect of value on behaviour

Value for the agents had positive and negative valence, as well as a degree of dynamics. There was a static negative value of being caught by a predator. Starvation depended on how much time was spent away from a food source. The positive value from food depended on how much energy was depleted, since the reward is the difference between 5.0 and the current energy level.

We wondered how value levels in the model affected behaviour, and if these effects were different when comparing a real to a simulated environment. Therefore, we looked at the relationship between energy level and behaviour. We looked at actions in two ways: expected value of actions, and the likelihood of choosing an

Table 2 Simulation behaviour under different model conditions

	Random	Planning certain	Planning uncertain	ActorCritic certain	ActorCritic uncertain
food	232.5 ± 44.3	494.9 ± 3.1	433.0 ± 7.9	325.1 ± 43.2	272.3 ± 43.4
starve	95.4 ± 23.5	3.0 ± 2.2	15.1 ± 6.9	73.1 ± 25.4	88.2 ± 29.42
caught	172.1 ± 23.9	2.1 ± 2.9	51.9 ± 3.2	101.8 ± 20.0	139.5 ± 22.2

Mean ± standard deviation of number of outcomes.

Table 3 Robot behaviour in the different layouts (planning vs. random model)

Layout A		Layout B		Layout C		Layout D		Layout E		Layout F	
Plan	Rand	Plan	Rand	Plan	Rand	Plan	Rand	Plan	Rand	Plan	Rand
F	S	F	C	F	S	F	F	F	S	F	S
F	S	C	C	F	C	F	S	F	S	C	S
F	S	F	C	F	S	C	F	F	S	F	S
F	S	F	C	C	F	C	F	F	F	C	S
F	F	C	F	F	F	F	F	F	F	F	S

C = Caught by predator, F = Found food, S = Starved.

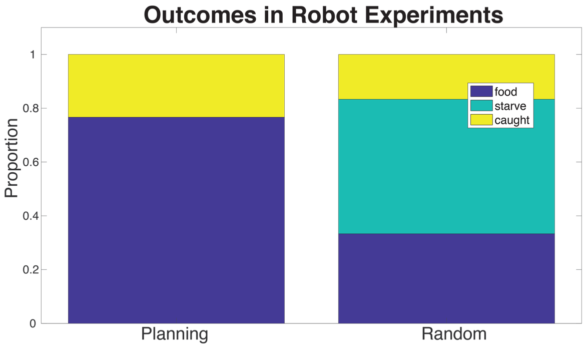

Fig. 8 *Behavioural outcomes in the robot experiments. For every trial, an outcome for the robot could be found food, starved, or caught by predator. The proportions of these three outcomes are shown in the figure. Robot experiments were conducted in 6 different layouts with 5 trials per layout*

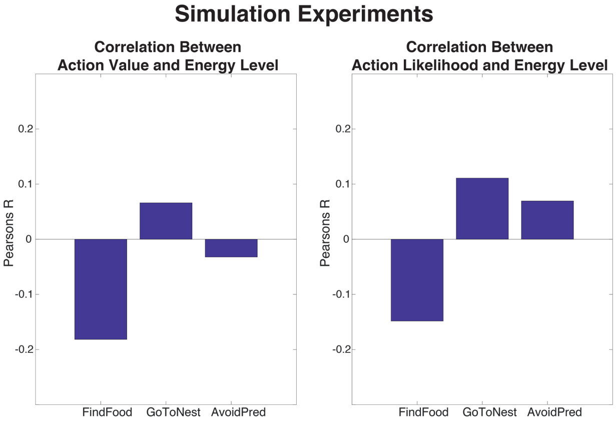

Fig. 9 *Relationship between actions and energy level for the Planning with Uncertainty agent in simulations. The bar graphs are based on correlations between 56,878 actions and energy level data points. Left. Correlations with the energy level are shown for the expected value of each potential action. Right. Correlations with the energy level are shown for the likelihood of taking an action. All correlations shown are significantly different from 0.0 (p < 0.00001)*

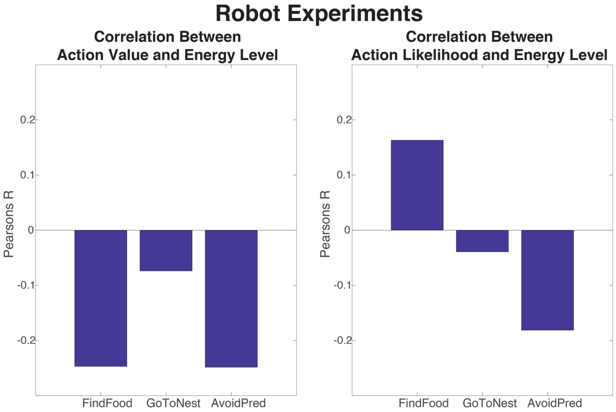

Fig. 10 *Relationship between actions and energy level for the Planning with Uncertainty agent in robot experiments. The bar graphs are based on correlations between 446 actions and energy level data points. The FindFood and AvoidPred correlations were significantly different from 0.0 (p < 0.00001). The GoToNest correlations were not statistically significant. Left. Correlations with the energy level are shown for the expected value of each potential action. Right. Correlations with the energy level are shown for the likelihood of taking an action*

action. In our Planning agents, the expected values for each were the Q values from (6), and the likelihood of those actions being taken were the probabilities of selecting each action, which was derived from (5).

In the simulations, there were strong negative correlations between finding food and energy level, which meant that the expected value and likelihood of seeking food increased when the agent's energy level was low (see Fig. 9). Although the expected value of avoiding a predator was not strongly correlated energy levels, the likelihood of avoiding a predator tended to be higher when energy levels were high. This is most likely due to the effect of the Softmax equation. When the expected value of food is high, the Softmax will amplify the likelihood of seeking food, and dampen the likelihood of going to the nest or avoiding a predator. It should be noted that small correlations (i.e. <0.1) were still statistically different from 0 due to the very large number of data points (i.e. 56,878) in the simulations.

Compared to the simulations, the value relationship differed in the robot experiments (see Fig. 10). The expected values of finding food and avoiding the predator were strongly negatively correlated with energy level. However, the likelihood of seeking food was positively correlated with energy level and the likelihood of avoiding a predator was negatively correlated with energy level. A more careful analysis of the data and observing the robots in action sheds light on this dichotomy. Compared to the simulations, the robot took more time to orient and move towards a goal. This may have given the predator more opportunities to locate and approach the agent. As the predator closes the gap between itself and the agent, the need to avoid the predator increases, at the expense of seeking food. In addition, by the time the predator chased the agent in earnest, the agent's energy level was low, and it tended to be on a path towards food. Oftentimes, a choice of avoiding the predator moved the agent closer to food in these situations. These observations further support the benefit of testing algorithms in embodied models.

4 Conclusion

In a set of simulation and robot experiments, we highlighted the advantage of planning ahead before taking actions in dynamic, noisy environments. This form of mental simulation predicted outcomes and overcame uncertainty. Indeed, mental imagery has been shown to influence future actions [21]. Mentalising, which is the ability to understand another's state and planning accordingly, is an important aspect of cognitive and social behaviour, and interestingly, mentalising is impaired in subjects with autism or autistic traits [27–29]. Although our Planning algorithm was simple, it was able to recursively ‘imagine’ how the effects of its own actions would affect another's actions. This is, in effect, a form of mental imagery or mental simulation. Mental simulation or imagery is thought to be an important component for developing artificial systems that are cognitive or conscious [22, 23]. However, such mental imagery can take time and can increase cognitive load.

Prediction is crucial for fitness in a complex world and a fundamental computation in cortical systems [16, 30, 31]. This requires the construction and maintenance of an internal model. Model-based reinforcement learning algorithms require the presence of an internal model and ability to represent transitions between states, which is the main difference from model-free reinforcement learning [32]. These algorithms have been used in predictive controllers for robots. For example, in a set of humanoid robot experiments, it was shown that having a proactive predictive model helped in predictive appropriate movements [33]. A combination of model based and model-free reinforcement learning was used in a sorting task [34]. The robot had to push cubes on a conveyor belt. The model-based system improved performance by maintaining a plan from one decision to the next. However, they suggested that the model-free system scales better under certain conditions and may be better in the face of uncertainty. Interestingly neural correlates of both model-based reinforcement learning and model-free reinforcement learning have been observed in the brain [25, 35]. In general, the brain maintains internal models for a wide range of behaviours; from motor control to language processing [17, 18].

In our experiments, the utilisation of an internal model was advantageous for predicting outcomes in uncertain conditions. The internal model not only maintained transitions between states, but also had a model of another's intentions. In both simulations and robot experiments, being able to mentalise sequences of actions led to better decisions. In simulations, the Actor Critic approach had difficulties handling uncertainty. It may be the case that our Planning approach was able to make better predictions by making multiple draws from distributions estimating object locations, similar to bootstrapping and sampling. Our Planning approach did not attempt to construct its internal model from scratch. The predator's actions and the safety of the nest were given, rather than learned. If these relationships changed, our present algorithm would not be able to adapt. In these cases, having a system that can fluidly switch between model based and model-free learning would be advantageous [33, 34], especially when conditions change, or circumstances require immediate action. In the future, it would be interesting to use this approach in a predator–prey scenario where the predator's strategy can change over time.

Our robot experiments highlight the importance of using embodied models to test cognitive hypotheses [36, 37]. Interesting strategies emerged in the robot trials that were due to interaction with the environment and the use of local sensing in the real world. For example, the agent robot's ability to locate objects was limited by the panoramic camera. Although this provided a 290° view, the reflection and warping led to poor resolution. In addition, objects were sometimes occluded (e.g. the predator sometimes blocked the view of the food, and vice versa). The Planning agent, by maintaining estimations of objects and predicting outcomes, was able to overcome this unreliability. It should be noted that because the state space was small, the robot planning agent could exhaust all possible outcomes in real-time. This may not be possible in a larger state space, especially if the robot must have onboard computation, as was the case in the present experiments. In a real-world setting, a fast-acting model-free algorithm, such as the Actor Critic, may have an advantage when there is time pressure to make a decision.

Noisy sensing was a factor in the predator robot's behaviour. The predator had a higher resolution image with its forward-facing camera, but a limited field of view. The robot agent could take advantage of this. At times, when the predator was moving randomly (e.g. when the agent was in the nest or out of view), the agent could get a head start towards a goal, such as food. Another interesting factor to emerge in the robot experiments was if the agent had a head start towards a goal, with the predator in pursuit, both finding food and avoiding predator actions led to the goal. Not all emergent properties were advantageous. A major difference between simulation and robot experiments was the physical properties of the robot's motor system, which used scrub steering tank treads. For the robot to take an action, such as find food, it needed to orient toward the goal and then move forward. During this time, the predator could close the gap between itself and the agent. Moreover, one incorrect decision could lead to the agent being caught, because physically correcting that action on the next decision takes time. It should be noted that these strategies and behaviours were not observed when the same model was used in the simulations. All of these observations provide support for using physical systems when designing and testing cognitive architectures [38].

Pertinent to the development of artificial cognitive systems, Karl Friston developed a thermodynamically motivated theory, which he called the ‘Free Energy Principle’, to describe brain processing and adaptive behaviour in biological organisms [1]. The underlying premise is that a biological system must maintain homeostasis to survive. To do so, it has to minimise its long-term average surprise, which relates to reducing the entropy experienced on its inputs. In our model, surprise was related to a value prediction error or unexpected uncertainty [3]. In the Planning agent, uncertainty due to sensor noise and occlusion (in the robot trials) led to surprise. In neuroscience, this implies that the brain constructs a model of the world in order to make predictions about sensory input and action outcomes, thus minimising surprise, and adapting when these predictions are erroneous. Agents, biological or otherwise, evolve a policy that minimises surprise by minimising the difference between likely and desired outcomes, which involves both pursuing the goal-state that has the highest expected utility (exploitation) and visiting a number of different goal-states (exploration). In this way, novelty seeking and curiosity reduces entropy in the long run [39]. Since dynamics are the result of energy flow, learning by prediction can be viewed as also learning the thermodynamics of the environment. For a system embedded within a dynamic ‘real’ world, knowledge of the physics of the environment is an absolute necessity, as every roboticist knows. In the context of the present work, our experiments were designed to underscore some of these ideas, and take a step towards the creation of a physically grounded cognitive system.

5 References

[1] Friston, K.: ‘The free-energy principle: a unified brain theory?’, *Nat. Rev. Neurosci.*, 2010, **11**, (2), pp. 127–138

[2] Ashby, W.R.: ‘Principles of the self-organizing dynamic system’, *J. Gen. Psychol.*, 1947, **37**, (2), pp. 125–128

[3] Yu, A.J., Dayan, P.: ‘Uncertainty, neuromodulation, and attention’, *Neuron*, 2005, **46**, (4), pp. 681–692 (in English)

[4] Edelman, G.M.: ‘Neural Darwinism: selection and reentrant signaling in higher brain function’, *Neuron*, 1993, **10**, (2), pp. 115–125

[5] Krichmar, J.L.: ‘The neuromodulatory system – a framework for survival and adaptive behavior in a challenging world’, *Adapt. Behav.*, 2008, **16**, pp. 385–399

[6] Lewis, M., Canamero, L.: ‘Hedonic quality or reward? A study of basic pleasure in homeostasis and decision making of a motivated autonomous robot’, *Adapt. Behav.*, 2016, **24**, (5), pp. 267–291 (in English)

[7] Lones, J., Canamero, L.: ‘Epigenetic adaptation through hormone modulation in autonomous robots’. 2013 IEEE Third Joint Int. Conf. on Development and Learning and Epigenetic Robotics (ICDL), Osaka, Japan, 2013 (in English)

[8] Krichmar, J.L.: ‘Neurorobotics – a thriving community and a promising pathway toward intelligent cognitive robots’, *Front. Neurorobot.*, 2018, **12**, (42), (in English), pp. 1–11

[9] Krichmar, J.L.: ‘Design principles for biologically inspired cognitive robotics’, *Biol. Inspired Cognit. Archit.*, 2012, **1**, pp. 73–81

[10] Krichmar, J.L., Edelman, G.M.: ‘Brain-based devices for the study of nervous systems and the development of intelligent machines’, *Artif. Life*, 2005, **11**, (1–2), pp. 63–78

[11] Pfeifer, R., Scheier, C.: ‘*Understanding intelligence (A Bradford book)*’ (MIT Press, 2001)

[12] Steels, L.: ‘When are robots intelligent autonomous agents?’, *Robot. Auton. Syst.*, 1995, **15**, (1), pp. 3–9

[13] Krakauer, J.W., Ghazanfar, A.A., Gomez-Marin, A.*, et al.*: ‘Neuroscience needs behavior: correcting a reductionist bias’, *Neuron*, 2017, **93**, (3), pp. 480–490

[14] Brooks, R.A.: ‘Intelligence without representation’, *Artif. Intell.*, 1991, **47**, (1–3), pp. 139–159 (in English)

[15] Brooks, R.A.: ‘Intelligence without reason’. presented at the Proc. 12th Int. Joint Conf. Artificial intelligence – volume 1, Sydney, New South Wales, Australia, 1991

[16] Clark, A.: ‘Whatever next? Predictive brains, situated agents, and the future of cognitive science’, *Behav. Brain Sci.*, 2013, **36**, (3), pp. 181–204

[17] Hickok, G., Houde, J., Rong, F.: ‘Sensorimotor integration in speech processing: computational basis and neural organization’, *Neuron*, 2011, **69**, (3), pp. 407–422

[18] Shadmehr, R., Krakauer, J.W.: ‘A computational neuroanatomy for motor control’, *Exp. Brain Res.*, 2008, **185**, (3), pp. 359–381 (in English)

[19] Craig, A.D.: ‘How do you feel – now? The anterior insula and human awareness’, *Nat. Rev. Neurosci.*, 2009, **10**, (1), pp. 59–70

[20] Critchley, H., Seth, A.: ‘Will studies of macaque insula reveal the neural mechanisms of self-awareness?’, *Neuron*, 2012, **74**, (3), pp. 423–426

[21] Pearson, J., Clifford, C.W., Tong, F.: ‘The functional impact of mental imagery on conscious perception’, *Curr. Biol.*, 2008, **18**, (13), pp. 982–986

[22] Chella, A., Manzotti, R.: ‘Artificial consciousness’, *Percept.-Action Cycle, Models Archit. Hardware*, 2011, **1**, pp. 637–671 (in English)

[23] Vernon, D., Metta, G., Sandini, G.: ‘A survey of artificial cognitive systems: implications for the autonomous development of mental capabilities in computational agents’, *IEEE Trans. Evol. Comput.*, 2007, **11**, (2), pp. 151–180 (in English)

[24] Foster, D.J., Morris, R.G., Dayan, P.: ‘A model of hippocampally dependent navigation, using the temporal difference learning rule’, *Hippocampus*, 2000, **10**, (1), pp. 1–16

[25] Glascher, J., Daw, N., Dayan, P.*, et al.*: ‘States versus rewards: dissociable neural prediction error signals underlying model-based and model-free reinforcement learning’, *Neuron*, 2010, **66**, (4), pp. 585–595

[26] Oros, N., Krichmar, J.L.: ‘Smartphone based robotics: powerful, flexible and inexpensive robots for hobbyists, educators, students and researchers’. CECS Technical Report 13–16, 2013, pp. 1–11

[27] Craig, A.B., Grossman, E., Krichmar, J.L.: ‘Investigation of autistic traits through strategic decision-making in games with adaptive agents’, *Sci. Rep.*, 2017, **7**, (1), p. 5533

[28] White, S., Hill, E., Happe, F.*, et al.*: ‘Revisiting the strange stories: revealing mentalizing impairments in autism’, *Child Dev.*, 2009, **80**, (4), pp. 1097–1117

[29] White, S.J., Frith, U., Rellecke, J.*, et al.*: ‘Autistic adolescents show atypical activation of the brain's mentalizing system even without a prior history of mentalizing problems’, *Neuropsychologia*, 2014, **56**, pp. 17–25

[30] George, D., Hawkins, J.: ‘Towards a mathematical theory of cortical micro-circuits’, *PLoS Comput. Biol.*, 2009, **5**, (10), p. e1000532

[31] Richert, M., Fisher, D., Piekniewski, F.*, et al.*: ‘Fundamental principles of cortical computation: unsupervised learning with prediction, compression and feedback’, arXiv:1608.06277, 2016

[32] Solway, A., Botvinick, M.M.: 'Goal-directed decision making as probabilistic inference: a computational framework and potential neural correlates', *Psychol. Rev.*, 2012, **119**, (1), pp. 120–154

[33] Murata, S., Arie, H., Ogata, T., *et al.*: 'Learning to generate proactive and reactive behavior using a dynamic neural network model with time-varying variance prediction mechanism', *Adv. Robot.*, 2014, **28**, (17), pp. 1189–1203 (in English)

[34] Renaudo, E., Girard, B., Chatila, R., *et al.*: 'Respective advantages and disadvantages of model-based and model-free reinforcement learning in a robotics neuro-inspired cognitive architecture', *Procedia Comput. Sci.*, 2015, **71**, (Suppl. C), pp. 178–184

[35] Daw, N.D., Gershman, S.J., Seymour, B., *et al.*: 'Model-based influences on humans' choices and striatal prediction errors', *Neuron*, Research Support, N.I.H., Extramural Research Support, Non-U.S. Gov't, 2011, **69**, (6), pp. 1204–1215 (in English)

[36] Pezzulo, G., Barsalou, L.W., Cangelosi, A., *et al.*: 'The mechanics of embodiment: a dialog on embodiment and computational modeling', *Front. Psychol.*, 2011, **2**, p. 5

[37] Pezzulo, G., Barsalou, L.W., Cangelosi, A., *et al.*: 'Computational grounded cognition: a new alliance between grounded cognition and computational modeling', *Front. Psychol.*, 2012, **3**, p. 612

[38] Pfeifer, R., Iida, F., Lungarella, M.: 'Cognition from the bottom up: on biological inspiration, body morphology, and soft materials', *Trends Cogn. Sci.*, 2014, **18**, (8), pp. 404–413

[39] Schwartenbeck, P., Fitzgerald, T., Dolan, R.J., *et al.*: 'Exploration, novelty, surprise, and free energy minimization', *Front. Psychol.*, 2013, **4**, p.710