Exploring Uncertainty and Movement in Categorical Perception using Robots

Nathaniel Powell, Josh Bongard

Department of Computer Science, University of Vermont, USA

Abstract. Cognitive agents are able to perform categorical perception through physical interaction (active categorical perception; ACP), or passively at a distance (distal categorical perception; DCP). It is possible that the former scaffolds the learning of the latter. However, it is unclear whether DCP indeed scaffolds ACP in humans and animals, nor how a robot could be trained to likewise learn DCP from ACP. Here we demonstrate a method for doing so which involves uncertainty: robots are trained to perform ACP when uncertain and DCP when certain. Furthermore, we demonstrate that robots trained in such a manner are more competent at categorizing novel objects than robots trained to categorize in other ways. This suggests that such a mechanism would also be useful for humans and animals, suggesting that they may be employing some version of this mechanism.
Keywords: Uncertainty; Active Categorical Perception; Robotics

1 Introduction

The embodied approach to cognitive science holds that the body is a necessary component for the acquisition of adaptive—and, ultimately, cognitive—behavior [3,7]. Since the establishment of this approach, much work has been dedicated to investigating how the body can do so [8], and quantifying its contribution [4,6]. A common approach for doing so is to employ robots, in which all aspects of their morphology, control structure, and task environment can be observed and experimentally modified.

A common skill investigated from an embodied perspective is categorical perception: how an agent makes use of its body to generate the requisite stimuli to learn appropriate categories. Initially, Beer evolved minimally cognitive agents to achieve this 'active' form of categorical perception (ACP) [1]: the agents interacted with their environment in a way that reduces intracategorical differences and magnifies intercategorical ones. Subsequent studies explored this phenomenon using more complex robot morphologies [2, 11, 12]; those robots physically manipulated the objects to be categorized. However, sophisticated cognitive agents typically employ distal categorical perception (DCP)—categorizing an object by sight and/or sound from a distance—as it has obvious advantages over ACP, such as rapidity, avoidance of potentially dangerous contact, and success even when physical contact is not possible. This raises questions regarding how animals learn (and how robots should learn) DCP, ACP, and how and when

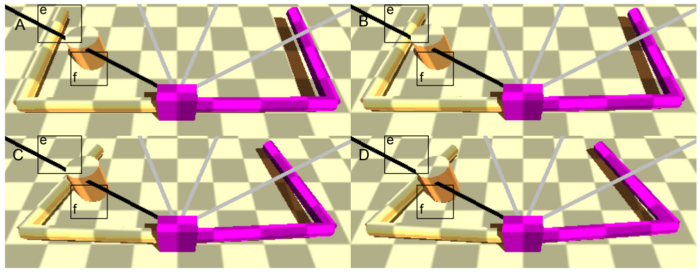


Fig. 1. Evolved behavior for a robot that interacts with its environment when uncertain to perform categorical perception. Time is tracked through panels A through D. Movement of the arm is tracked in box e. Movement of the object is tracked in box f.

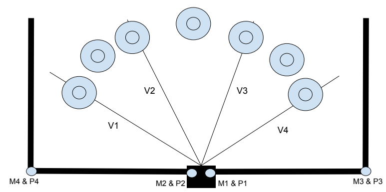
to switch between them. One hypothesis is that ACP scaffolds the learning of DCP: interactions with objects can structure perception in such a way as to facilitate learning of non-embodied skills [5].

The question remains however as to the conditions under which ACP or DCP should be employed. We hypothesize that such switching should be modulated by uncertainty: unfamiliar stimuli should trigger internal uncertainty, which in turn should trigger appropriate action resulting in ACP, which, finally, provides scaffolding for the learning of DCP when next presented with this object. Over a lifetime, this should result in an agent that exhibits more instances of DCP and fewer of ACP. Here, we demonstrate the usefulness of this particular mechanism by training simulated robots to perform ACP when uncertain and DCP when certain. We show that, when exposed to novel stimuli, these robots categorize better than robots trained to categorize in other ways.

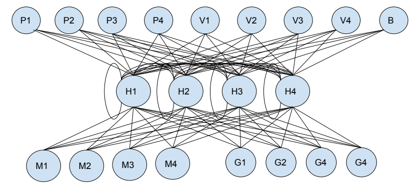
2 Methods

We conducted a series of experiments in which a simulated yet embodied robot attempts to categorize objects in its environment (Fig. 1). It is embodied in the sense that, despite its virtual surroundings, actions it performs can impact the environment, and the agent immediately detects the sensory repercussions of those effects. An evolutionary algorithm was used to train the robot to perform ACP, DCP, or a combination of the two when exposed to a number of environments. The robot's task was to correctly categorize large objects as large, and small objects as small. When training concludes, the best robot's categorization abilities were tested by exposing it to novel environments, and its categorization error in those novel situations was measured.

The robot body. The robot's body was constructed from four equal length cylinders and a small central body constructed from a rectangular solid. The arms adjacent to the main body are connected by supporting motorized joints to it, as are the forearms connected to the upper arms (Fig. 2a). There are a total



a



b

Fig. 2. (a) The robot body is constructed from two arms. The upper limbs are attached to a central body. The limbs are attached to each other and the body with four motorized joint, each containing an angle sensor. Thin lines represent line of sight for sensing distance. The robot can be exposed to any one of 14 objects, which are placed either in the line of sight or 'blind' to the robot. Objects are either large (large circles) or small (small circles). (b) The controller of the robot is instantiated as an artificial neural network. The input layer is made up of four proprioceptive neurons (P), four vision neurons (V) and a bias neuron (B). Hidden layer consists of 4 recurrent hidden neurons (H). Output layer: four motor neurons (M) and four guess neurons (G)

of four motorized joints, yielding four mechanical degrees of freedom. Each of the motorized joints enabled the connected objects to rotate relative to one another through the robot's coronal plane, which, given its morphology, corresponds to the horizontal plane. This results in the arms flexing in horizontally toward the main body and extending outward from the main body, also horizontally. The arms were placed at a particular height so that they do not end up breaking the visual beams emanating from the robot (black and gray lines in Fig. 1).

The robot controller. The controller of the robot is instantiated as a partially-recurrent artificial neural network. There are three layers that make up the neural network: the input layer, a hidden layer, and an output layer (Fig. 2b). The input layer consists of two types of sensory neurons: vision neurons (V) and proprioceptive neurons (P). At each time step during which a robot is simulated, the angles of the four motorized joints are computed, normalized to real values in the range $[-1, +1]$, and supplied to the four proprioceptive neurons. Likewise, four beams are sent out from the main body such that they span the range $[-60^o, +60^o]$ in front of the robot. The angles between each pair of neighboring beams was set to 40^o. While the robot may move its arms, it cannot move the visual beams. However, the beams may be broken by coming into contact with an external object. The length of each beam at each time step is computed and scaled to a real value in $[0, 1]$ such that zero indicates the beam is unbroken, while one indicates that an object is in contact with the base of the beam. The hidden layer consists of four fully recurrent hidden neurons (H): each hidden neuron receives input from each of the sensors (in addition to a fixed-output bias neuron B) as well as values from the other hidden neurons, including itself.

The new value of the ith hidden neuron h_i is computed as

$$h_i = \tanh((\sum_{j=1}^{4} s_j w_{ji}) + w_{bi} + (\sum_{k=1}^{4} h_k w_{ki})) \quad (1)$$

where s_j is the value of the jth sensor neuron, w_{ji} is the weight of the synapse connecting the jth sensor neuron to the ith hidden neuron ($w_{ji} \in [-1, +1]$), w_{bi} is the weight of the synapse connecting the bias neuron (value clamped to one) to the ith hidden neuron, h_k is the value of the kth hidden neuron, w_{ki} is the weight of the synapse connecting the kth hidden neuron to the ith hidden neuron, and $\tanh(x)$ brings the hidden neuron values back into the range $[-1, 1]$.

The output layer is comprised of two different types of neurons: motor neurons (M) and guess neurons (G). The value of the ith output neuron is computed at each time step using $o_i = \tanh(\sum_{j=1}^{4} h_j w_{ji})$ regardless of whether it is a motor or guess neuron. The value of each of the four motor neurons is scaled to the range $[-45^o, +45^o]$ and then supplied, as a desired angle, to each of the four joints. A proportional-derivative (PD) controller is effected by supplying torque to the joint proportional to the difference between the current angle and the desired angle. The outputs arriving at the guess neurons are employed by the robot to perform categorical perception and are described in more detail below.

The task environment. When evaluated, a robot is equipped with a neural network labeled with a particular set of synaptic weights as described above. The robot is then exposed to one of 14 environments as shown in Fig. 2. There are seven possible positions. At each location there are two possible types of objects that may appear: a cylinder with a large or small radius. Objects are placed in such a way that they are either initially unseen by the robot or in its direct line of sight.

The evolutionary algorithm. An evolutionary algorithm was used to train neural networks in the robot as described above. For each evolutionary trial, seven environments from the total set of 14 were chosen as random and fixed as the training set for that trial. In a different evolutionary trial, seven different environments may be chosen. At the outset of the trial, an initial population of 20 random neural networks were created. Each of these neural networks contained random synaptic weights drawn from $[-1, +1]$ with a uniform distribution. Each neural network was then evaluated on the robot seven times, in the seven environments chosen for that trial. During each of these seven evaluation periods, the robot was allowed to move for 25 time steps in the simulator. This population of neural networks was then evolved for 100 generations using a common evolutionary algorithm that balances increasing fitness over time while also maintaining genetic diversity in the population [9]. When the 100 generations complete, the ANN with highest fitness in the population is re-evaluated on the robot seven times, in the seven novel environments that were unseen during evolution.

The fitness functions. The robots in this experiment were evaluated against four different fitness functions, leading to four experimental conditions. In the first condition, robots were evolved simply to categorize correctly (C). In the second condition they were evolved to categorize correctly while minimizing movement (CnM). In the third condition they were evolved to categorize correctly while maximizing movement (CM). In the fourth and final condition they were evolved to categorize correctly, and to do so by moving when uncertain about the object's category and remaining still when certain about the object's category. This condition was referred to as CR, as the robot should establish a correlation (R) between movement and uncertainty. Fifteen independent evolutionary trials, each starting with a different randomly chosen set of seven objects and 20 random ANNs, were performed for each of the four conditions, yielding a total of 60 independent trials.

The fitness functions for these conditions were constructed from combinations of the following terms

$$C = \sum_{t=1}^{T}\sum_{e=1}^{E}\sum_{i=1}^{G}(g_{ei}^{(t)} - s_e)^2/TEG \tag{2}$$

$$K = \sum_{t=1}^{T}\sum_{e=1}^{E}\sum_{m=1}^{M}|(a_{em}^{(t)} - a_{em}^{(t-1)})|/TEM \tag{3}$$

$$R = \sum_{e=1}^{E}\mathrm{Corr}(\boldsymbol{G}_e, \boldsymbol{m}_e)/E \tag{4}$$

$$\boldsymbol{G}_e = [\sigma(\boldsymbol{g}_e^{(2)}), \sigma(\boldsymbol{g}_e^{(3)}), \dots, \sigma(\boldsymbol{g}_e^{(25)})] \tag{5}$$

$$\boldsymbol{m}_e = [\sum_{m=1}^{M}|a_{em}^{(2)} - a_{em}^{(1)}|, \dots, \sum_{m=1}^{M}|a_{em}^{(25)} - a_{em}^{(24)}|] \tag{6}$$

where

- C denotes how well a given neural network categorizes, averaged over all $T = 25$ time steps, E=7 training objects, and G=4 guess neurons ($C = 0$ indicates perfect categorization and $C = 1$ the worst possible categorization);
- K denotes the average amount of motion over all T time steps, E training environments, and $M = 4$ motors;
- R denotes the amount of correlation (Corr) between uncertainty ($\boldsymbol{G}_e$) and amount of movement ($\boldsymbol{m}_e$), averaged over all E environments ($R = 1$ indicates the robot moves maximally when uncertain and minimally when certain);
- $\boldsymbol{G}_e$ represents a vector containing the uncertainties of the ANN when exposed to the eth environment, at each time step of the exposure (with the exception of the first time step);
- $\boldsymbol{m}_e$ represents a vector containing the amount that the robot moved when exposed to the eth environment, at each time step of the exposure (with the exception of the first time step);

- $g_{ei}^{(t)}$ represents the output of the ith guess neuron during the tth time step of exposure to the oth object;
- s_e represents the size of the object in the eth environment (small object=-0.5, large object=+0.5);
- $a_{em}^{(t)}$ represents the angle of the mth motorized joint during the tth time step of exposure to the eth environment; and
- $\boldsymbol{g}_e^{(t)}$ represents a vector containing the values of the four guess neurons generated during the tth time step when exposed to the eth environment, and $\sigma(\boldsymbol{g}_e^{(2)})$ represents the variance within that vector.

Condition *C*: In the first condition, robots were only evolved to categorize, regardless of the amount or type of movement they employed to do so. This was accomplished by evolving robots that maximized the fitness function

$$F_C = 1/(1 + C). \tag{7}$$

Condition *CnM*: In the second condition, DCP was explicitly favored by evolving robots that successfully categorize while also minimizing movement:

$$F_{CnM} = (\frac{1}{1 + C})(\frac{1}{1 + K}). \tag{8}$$

Condition *CM*: In this third condition, ACP was explicitly favored by evolving robots to successfully categorize while maximizing movement:

$$F_{CM} = K/(1 + C). \tag{9}$$

Condition *CR*: Finally, robots were evolved in the fourth condition to employ DCP when uncertain as to the object's size and to employ ACP when they were certain. This was accomplished using

$$F_{CR} = R/(1 + C). \tag{10}$$

Prediction variance and uncertainty. Here, we employ the variance among the values of the guess neurons to denote a controller's uncertainty about the current object's category. In the machine learning literature, prediction variance is often employed as a proxy for uncertainty [10]. This is because, as long as individual units in a predictive model (here, the guess neurons) are independent, they are likely only to converge on the same prediction when that prediction is correct. This is not unlike a group of people with very different backgrounds generating diverse—and thus mostly wrong—answers to questions that touch on an area of their mutual ignorance, but who only generate similar responses when the question touches on an area of their common knowledge. The guess neurons here are independent because each guess neuron has its own synaptic weights connecting it to the input layer.

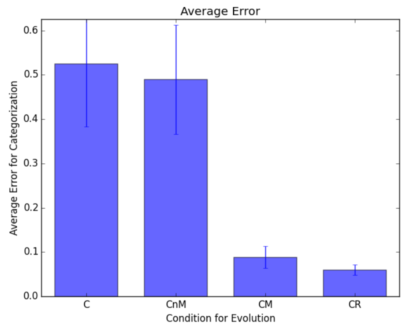


Fig. 3. The average ability of the best 15 controllers to correctly categorize unseen objects, for the four experimental conditions tested. The bars are in reference to the standard error for each condition.

3 Results

At the termination of each run, for each condition, the robot with the best fitness is extracted from the population. Each of these controllers is then re-instantiated in the robot and evaluated a further seven times, in the seven environments that the controller did not experience during training. We employed C (Eqn. 2) to compute the robot's average ability to categorize these $E = 7$ novel environments. A controller that obtains lower values of C when exposed to these novel environments is thus exhibiting a better ability to generalize its ability to categorize, compared to another controller with a higher value of C. Fig. 3 reports the average generalization abilities of the 15 evolved controllers extracted from the four conditions. Testing for significance was computed using multiple Mann-Whitney U tests. The tests looked for significant differences between the CR condition and the remaining three conditions (C, CnM, and CM). Significant P-values were found in each comparison. After correcting for multiple comparisons using the Bonferroni method of adjustment, the P-values found between CR and C, CR and CnM, and CR and CM are respectively 1.52×10^{-5}, 1.52×10^{-5}, and 1.18×10^{-3}. Significant P-values indicate that the relative average amount of categorization error performed by individuals in the CR condition is significantly lower than the error of individuals in the remaining three conditions. Therefore, it can be concluded that individuals evolved under the CR condition are better suited for tasks involving categorization of objects than individuals from the other three evolutionary conditions.

4 Discussion

Why is *C* worse than *CR*? The controllers evolved in the C condition performed worse than those evolved for CR for several reasons. Such controllers may have generated little to no motion, enabling rapid and successful categorization of seen objects, while sacrificing the ability to categorize unseen objects, in the training environments. This may have led to overfitting such that novel, seen objects in the testing environments were categorized incorrectly. Conversely, controllers may have evolved to perform more motion. Such a strategy might cause correct, instantaneous categorization which is subsequently lost when the robot comes into contact with the object.

Why is CnM worse than CR? Controllers evolved in the CnM condition are likely incentivized to memorize the categories of seen objects, and ignore the categories of unseen objects. This results in overfitting: the robot will not only poorly categorize novel, unseen objects, but novel, seen objects as well. In other words, the robots are deprived of the ability to reduce spurious differences between intracategory objects through motion.

Why is CM worse than CR? Conversely, controllers evolved in the CM condition may suffer from two disadvantages compared to controllers evolved using CR. Controllers from CM may not be able to afford to hold still when they are certain of a novel object's class. Moreover, they may have to exhibit so much motion that they end up magnifying spurious intracategory differences, rather than being free to generate less—yet appropriate—movement that reduces those differences. A possible example of one (or both) of these disadvantages is shown in Fig. 4. Even though this controller can reduce category error when in the presence of a visible novel object by perhaps moving in a way that does not move this object (Fig. 4a), error increases (and then reduces to the original high level) when exposed to a novel blind object (Fig. 4b).

How did CR succeed? Controllers produced in the CR condition presumably outperformed the controllers produced by the other three conditions because they can better employ ACP, or DCP, when that form of categorization is most appropriate. When the robot is certain about an object's category, it should employ DCP without moving: the robot does not need to wait for physical contact with the object to categorize the object. Moreover, if it does contact the object, the resulting sensor data may affect the guess neurons and thus draw the robot away from the already correctly-predicted category. Conversely, when the robot is uncertain as to the object's category, it should initiate movement as rapidly as possible: that is, exactly at the moment that it is uncertain. Another interesting result from the CR condition is that when a CR-generated controller's uncertainty is high, it also tends to exhibit higher category error (exemplified by the spike in error and uncertainty in Fig. 4c). At such times, CR-generated controllers are encouraged to move as much as possible. However, once in contact with the object and the amount of uncertainty starts to decrease, less motion is necessary: the robot is free to perform whatever actions are appropriate to reduce intracategory differences. In contrast, CM generated robots are more restricted in the kinds of actions they can employ to reduce these differences: they must generate high-magnitude movements.

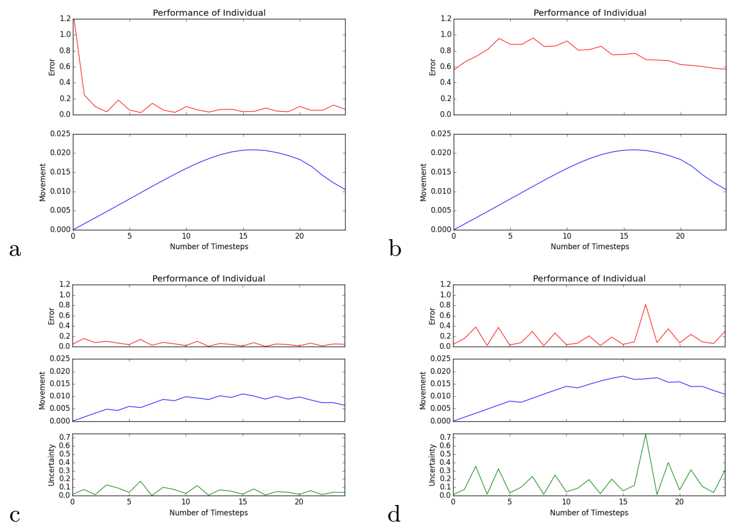


Fig. 4. Performance of a robot evolved to categorize and maximize movement (top). Average categorization error is seen in the top subplot, average change in joint angles is seen in the middle subplot. Performance of robot in categorizing a (a) visible novel object and (b) a blind novel object. Performance of a robot evolved with both C and R to categorize (bottom). Average categorization error is seen in the top subplot, average change in joint angles is seen in the middle subplot ,and the variance of the guess neurons is seen in the bottom subplot. Performance of robot in categorizing a (c) visible novel object and (d) a blind novel object.

Possible Sources of Error. One potential source of error is that only 100 generations of evolution were performed with a relatively small initial population of 20 individuals. This may not have allowed for significant optimization to occur in all conditions. The very short evaluation period may also be a source of error, because the CM and C conditions may allow for the discovery of exaggerated movements that yet, given enough time, eventually reduce intracategory differences. More generally, longer evaluation periods will allow for a greater range of movements that may then better help clarify the relationship between movement, categorization, and uncertainty.

5 Conclusion

Here we have shown that it is possible to explicitly train robots to exhibit ACP when uncertain and DCP when certain, and that such robots outperform other robots trained to perform ACP or DCP at will but without the ability to do so based on uncertainty; trained to always perform DCP (categorize without moving); or trained to always perform ACP (categorize via movement). Further, this approach does not require us to dictate how the robot should interact with its environment; it is free to discover its own strategies for reducing intracategory differences through physical interaction. This work helps to clarify the relationship between three competencies necessary for any embodied agent that wishes to categorize rapidly and successfully: the ability to categorize, the ability to interact with the world, and the ability to decide when such interactions are and are not needed for categorization. In future work we wish to employ more sophisticated optimization methods, such as multiobjective optimization, which enable better tradeoffs between multiple fitness terms. We would also like to investigate the kinds of physical interactions generated by the controllers, and how such actions differ based on different circumstances. Further, we wish to investigate how the successful controllers described here gradually acquired DCP, ACP, and/or whether the acquisition of one scaffolded the subsequent acquisition of the other. Finally, we wish to explore whether such dynamics relate to how humans categorize.

Acknowledgments. This work was supported by the National Science Foundation under projects PECASE-0953837 and INSPIRE-1344227.

References

1. Randall D Beer. The dynamics of active categorical perception in an evolved model agent. *Adaptive Behavior*, 11(4):209–243, 2003.
2. Josh Bongard. The utility of evolving simulated robot morphology increases with task complexity for object manipulation. *Artificial life*, 16(3):201–223, 2010.
3. Rodney A Brooks. Elephants don't play chess. *Robotics and Autonomous Systems*, 6(1):3–15, 1990.
4. Helmut Hauser, Auke J Ijspeert, Rudolf M Füchslin, Rolf Pfeifer, and Wolfgang Maass. Towards a theoretical foundation for morphological computation with compliant bodies. *Biological Cybernetics*, 105(5-6):355–370, 2011.
5. Max Lungarella and Olaf Sporns. Information self-structuring: Key principle for learning and development. In *Procs of the Intl Conf on Development and Learning*, pages 25–30. IEEE, 2005.
6. Chandana Paul. Morphological computation: A basis for the analysis of morphology and control requirements. *Robotics and Autonomous Systems*, 54(8):619–630, 2006.
7. Rolf Pfeifer and Josh Bongard. *How the Body Shapes the Way We Think: A New View of Intelligence*. MIT press, 2006.
8. Rolf Pfeifer, Fumiya Iida, and Max Lungarella. Cognition from the bottom up: on biological inspiration, body morphology, and soft materials. *Trends in Cognitive Sciences*, 18(8):404–413, 2014.
9. Michael Schmidt and Hod Lipson. Age-fitness pareto optimization. In *Genetic Programming Theory and Practice VIII*, pages 129–146. Springer, 2011.
10. H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Procs of the fifth annual workshop on computational learning theory*, pages 287–294. ACM, 1992.
11. Elio Tuci, Gianluca Massera, and Stefano Nolfi. Active categorical perception of object shapes in a simulated anthropomorphic robotic arm. *IEEE Transactions on Evolutionary Computation*, 14(6):885–899, 2010.
12. Karol Zieba and Josh Bongard. An embodied approach for evolving robust visual classifiers. In *Procs of the Genetic and Evolutionary Computation Conf*, pages 201–208. ACM, 2015.