An Embodied Approach for Evolving Robust Visual Classifiers

Karol Zieba
University of Vermont
Department of Computer Science
Burlington, Vermont 05401
kzieba@uvm.edu

Josh Bongard
University of Vermont
Department of Computer Science
Burlington, Vermont 05401
jbongard@uvm.edu

ABSTRACT

Despite recent demonstrations that deep learning methods can successfully recognize and categorize objects using high dimensional visual input, other recent work has shown that these methods can fail when presented with novel input. However, a robot that is free to interact with objects should be able to reduce spurious differences between objects belonging to the same class through motion and thus reduce the likelihood of overfitting. Here we demonstrate a robot that achieves more robust categorization when it evolves to use proprioceptive sensors and is then trained to rely increasingly on vision, compared to a similar robot that is trained to categorize only with visual sensors. This work thus suggests that embodied methods may help scaffold the eventual achievement of robust visual classification.

Categories and Subject Descriptors

I.2.m.c [**Artificial Intelligence**]: Evolutionary computing and genetic algorithms; I.2.9 [**Computing Methodologies**]: Artificial Intelligence - Robotics

Keywords

Evolutionary Computation; fitness; deception; scaffolding

1. INTRODUCTION

Categorization is an important aspect of intelligence [12], but fundamental disagreement exists as to how an artificial agent should do so, and how biological organisms acquire this ability.

One can partition categorization strategies into non-embodied and embodied approaches. In the non-embodied approach, an agent is presented with some stimuli and must signal which category the perceived object belongs to. In the embodied approach, the robot or organism must interact with its environment to generate useful percepts for categorization [1, 3, 22].

The disagreement about which of these two approaches to categorization is superior stems from the fact that humans are equally adept at both: one can nearly instantaneously visually recognize a friend at a distance or rapidly pick out a desired key from one's pocket just by handling a set of them.

Deep learning methods have recently demonstrated an excellent ability to recognize objects belonging to familiar categories using the non-embodied approach [2, 13]. These methods are able to handle input images with very high dimensionality because they are provided with millions of training images. However, despite these successes, recent work [21, 16] has demonstrated that these methods can fail on images that, from a human observer's perspective, clearly do not contain the object claimed to exist in the image.

Embodied approaches to categorization offer an advantage over non-embodied approaches in that the learner may choose how to manipulate objects such that spurious differences between objects in the same class, including orientation and position, are reduced. In addition, the learner may act to increase the differences between objects belonging to different classes: if edged objects are to be distinguished from round objects, the learner may alter her grasp of an object to confirm the existence (or absence) of an edge. It has even been shown that, given an appropriate morphology, a robot may reduce intra-category differences and increase inter-category differences as a side effect of actions that are not directed towards explicit object manipulation [20].

Evolutionary algorithms have been employed previously to enable a robot to perform this active categorical perception (ACP). Beer [1] reported an agent that achieves ACP simply by moving relative to objects in its environment without touching them, while Tuci [22] and Bongard [3] reported robots that achieved ACP by physically manipulating objects. Furthermore, Bongard [3] demonstrated that evolving robot morphology along with control facilitated the evolution of ACP, presumably because evolution could more readily discover grasping strategies that reduced intra-category differences and exaggerated inter-category differences. However, to date, no evolutionary approaches have show that tactile experiences predispose certain strategies to be robust in novel situations.

Outside of evolutionary robotics, Fitzpatrick *et al.* [10] presented work in which robots learn to visually classify objects based on their physical interactions with them. However, the robots were pre-programmed to explicitly detect correlations between proprioceptive and visual features. Here we describe a similar approach, but do not require that the

robot detect similarities between different sensor modalities. Instead, we employ scaffolding to gradually wean a robot off sensors that require physical contact and onto visual sensors that do not.

Scaffolding, a concept brought to robotics from developmental psychology [18], facilitates learning by initially exposing the learner to simpler tasks, and only exposing her to more challenging versions of the tasks gradually. Our use of scaffolding to swap one sensor modality in for another differs from most usages of scaffolding in robotics, in which the robot is exposed to increasingly more challenging task environments [8, 17, 19], or in which the robot's morphology itself scaffolds the acquisition of adaptive behavior [5, 4].

Here, we demonstrate a robot evolved to achieve active categorical perception using action and proprioception, which successfully reduces spurious intra-category differences. During subsequent evolution, these robots are challenged to rely increasingly on vision and allowed to rely less on proprioception, which gradually transitions the robot from ACP to visual classification. We demonstrate that the resulting robots retain the action that reduced intra-category differences and thus exhibit robust visual classification when exposed to novel visual scenes.

The next section describes this method in more detail. Sect. 3 reports our results, and the final section provides some discussion and concluding remarks.

2. METHODS

We first describe the task for our robot. We proceed to describe the robot's body and controller architecture. We then describe the robot's sensor modalities. This is followed by a description of how scaffolding is employed to wean categorizing robots off proprioception and force them to rely increasingly on vision alone. We also elaborate on the various environments the robots were trained in. We conclude this section by describing how we measured the robustness of the evolved robots when forced to categorize in previously unseen environments. All material for replicating the work described here is available at http://git.io/vfYYP

2.1 Task

The robot we are evaluating is tasked with classifying the size of a cylinder within its grasp. Two cylinder sizes are presented to each robot. These cylinders vary in their radius: the larger one's radius is 50% larger than the smaller one. The larger cylinder's radius was 30% of each of the robot's arm segments in length.

2.2 Robot Morphology

The robot's morphology (Figure 1) is planar and is comprised of five body segments connected together with four, one degree-of-freedom hinge joints. The bulk of the robot is comprised of its chassis, which is locked in place for the present study. The two arms are each connected to the chassis at slightly different heights to allow them to slide past each other if their grip flexes sufficiently far inward. Each arm is composed of an upper and lower segment. These segments are attached with a hinge joint that rotates the two arm segments through the horizontal plane, with a range of motion constrained to $[-90^o, +90^o]$. The upper segment is attached to the chassis with a second hinge joint that rotates the entire arm relative to the chassis through the range $[-90^o, +90^o]$. The initial pose of the robot, as shown at the top of Figure 1, is considered to set the four joint angles to default values of 0°.
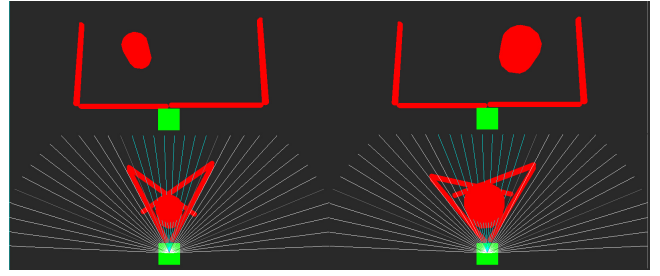

Figure 1: Each of the four frames show the robot under different environments. The top frames depict the start of simulations with a small object and a large object respectively. The bottom frames exhibit the rays the robot uses to see objects after the robot has gripped the objects during its simulation, with the slightly darker ray depicting the center of each eye.

Each of the four joints are equipped with a motor that applies a torque to the joint proportional to the difference between the joint's current angle and the desired angle output by the robot's controller. The robot is equipped with four proprioceptive sensors, which report the current angle of each joint.

Vision is, in the most fundamental sense, an instantaneous perception of remote objects. For this experiment we chose not to simulate vision, but rather to simulate a simpler set of distance sensors. Distance sensors operate much like visual ones, but instead of detecting variations in colors they detect variations in distance. Furthermore, like vision, distal sensors can be high resolution. Vision here is thus approximated using four sets of 'eyes', which point at -67.5^o, -22.5^o, $+22.5^o$, and $+67.5^o$ relative to the forward facing direction, arbitrarily considered to be 0°.

Each eye is composed of a fan of nine rays equally spaced (5°) apart. At each time step a cast ray returns a value linearly proportional to the distance between the source of the ray and the first point of collision. A maximum value is returned if the ray is unobstructed. The rays' values are then averaged and normalized to provide four visual inputs to the controller. A visual input value of -1 indicates a large object right in front of the sensor while $+1$ indicates there is no object within range of that eye. A higher resolution of rays was not used due to the linearly increasing computational cost of casting rays.

The following equation shows the setup of each of they vision sensors. The term N is the number of rays. The term R is the length of each of the rays. The subscript o refers to the origin of the ray and the subscript c refers to the point of first collision.

$$
\begin{aligned}
d_r &= \sqrt{(x_{r,o} - x_{r,c})^2 + (y_{r,o} - y_{r,c})^2 + (z_{r,o} - z_{r,c})^2} \\
v &= \frac{1}{N} \sum_{r=1}^{N} \begin{cases} 2\frac{d_r}{R} - 1 & \text{if ray } r \text{ collides} \\ 1 & \text{otherwise} \end{cases}
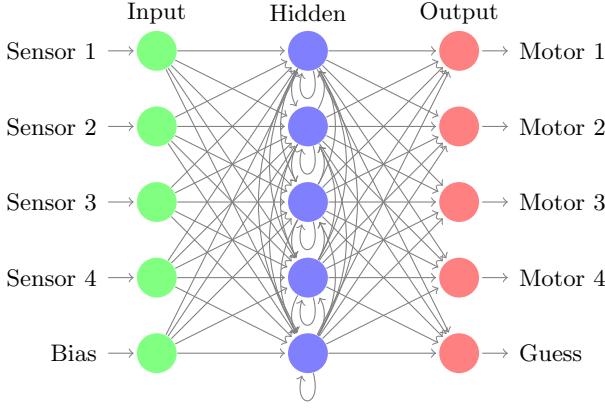\end{aligned} \tag{1}
$$

Figure 2: Neural Network of Controller

2.3 Controller

The robot's controller is a synchronous, deterministic, and real-valued neural network. Figure 2 reports its architecture, where each layer is fully connected to the succeeding layer. The middle (hidden) layer is also fully recurrent, obtaining inputs from all five input and five hidden neurons. The output layer's five neurons feed directly from the hidden layer. Four of the five input neurons were designated as sensor inputs. The fifth input neuron was a bias neuron permanently set to the maximum neuron value of one. Four of the five output neurons were used to control the joint motors. The final output neuron is the guess neuron, which was used for object categorization but did not influence the motion of the robot. At each time step the input neurons were encoded with the current sensor values. Each hidden neuron was then updated using:

$$h_i^{(t+1)} \;=\; \mathrm{erf}\left(\sum_{j=1}^{5} n_j^{(t+1)} w_{j,i} + \sum_{j=1}^{5} h_j^{(t)} w_{j,i}\right) \quad (2)$$

where n_j and h_j are the jth input and hidden neurons, respectively, $w_{j,i}$ is the synaptic weight connecting neuron j to neuron i, and this weighted sum is normalized to a value in $[-1, +1]$ using the Gauss error function. Synaptic weights were restricted to the range $[-1, +1]$.

The output neurons were updated using:

$$o_i^{(t+1)} \;=\; \mathrm{erf}\left(\sum_{j=1}^{5} h_j^{(t+1)} w_{j,i}\right) \quad (3)$$

After the network was updated, the values of the four motor neurons were scaled to values in $[-90^o, +90^o]$ and then translated into torques by the motors, proportional to how far the current angle was from the desired angle.

During the evolutionary runs in which the robot is weaned off proprioception and on to vision, some mixture of proprioception and vision is supplied to the sensor neurons, rather than feeding increasingly less proprioception to four sensor neurons and increasingly more vision to an additional four sensor neurons. In this way evolution does not need to learn to ignore or value sets of weights over the evolutionary run.

2.4 Evolutionary Algorithm

The Covariance Matrix Adaptation Evolution Strategy [11] (CMA-ES) was chosen as the real-valued optimization method. In all evolutionary trials, only the synaptic weights in the robot's controller were evolved. All aspects of the robot's cognitive architecture and morphology remained fixed. CMA-ES evolved 75 synaptic weights, each constrained to $[-1, 1]$. The initial synapses vector was initialized with uniformly random weights in the bounded range. CMA-ES specific parameters included initializing each run with a σ of 0.25 and a default of 16 fitness evaluations per generation. Each evaluation was composed of multiple simulations in which the robot was confronted with different objects placed at different positions. All other unmentioned parameters were kept at the default settings. Since we utilized CMA-ES as a function minimizer, our experiment attempted to minimize the error of the robot's guess as to which class the object currently in front of it belonged to. We shall use the term evolutionary run to refer to the process of evolving (training) our controllers for a given set of environments and sensor modality.

2.5 Environments

The environment of the robots differ primarily through the position and size of the cylinder presented in each simulation. Each robot's controller for a given sensor modality was simulated a specific number of times, which we define as an evaluation. During training the objects were placed as described below and shown in Figure 3 for each 6-simulation evaluation. The horizontal and vertical environments were chosen because they constrained the training data to one dimension. The alternating environment was chosen because it did not place both a large and a small cylinder at the same positions. Additionally, we also investigated how controllers evolved when exposed to fewer (4) and more (8) simulations. The evaluation types include:

Horizontal (H4, H6, H8) The objects were placed across the X axis such that both sizes were tried at each unique position.

Vertical (V4, V6, V8) The objects were placed across the Z axis such that both sizes were tried at each unique position.

Alternating (A4, A6, A8) The objects were placed in a two-object deep rectangle, alternating large and small objects, each object with their own unique position.

Testing The objects were placed on a Cartesian plane over 78 positions for a total of 156 simulations.

2.6 Sensor Modalities and Scaffolding

2.6.1 Proprioception (P)

Robots evaluated under this sensor modality only utilized their proprioceptive sensors (joint angles) as inputs to their controller for the entirety of training and testing.

2.6.2 Vision (V)

Robots evaluated under this sensor modality only utilized their vision sensors (four eyes composed of distal rays) as inputs to their controller for the entirety of training and testing.

2.6.3 Scaffolding

Although scaffolding is a common method employed in robotics [8, 17, 19, 5, 4], we employed it here in a novel way.
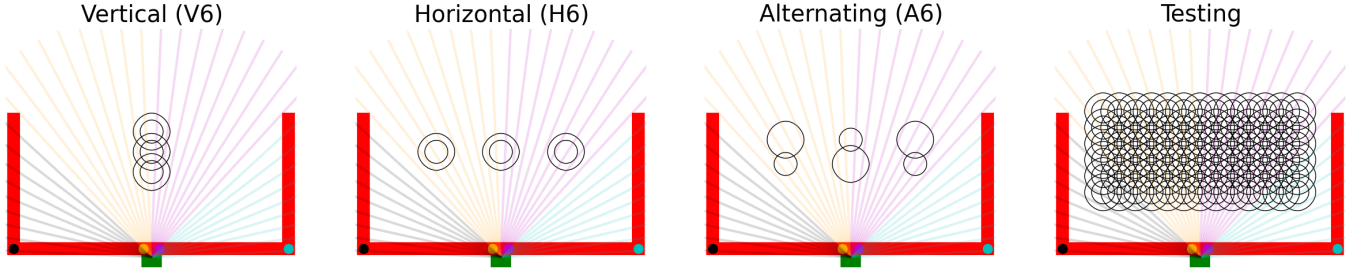
Figure 3: This figure depicts the initial positions for the cylinders under various environments. Additionally, the thick red lines indicate the initial position of the robot's limbs. The filled circles represent the locations of the joints. The circle and ray colors correspond to the ordered pairs of sensors combined during the ontological scaffolding: the teal outer joint on the right arm corresponds to the rightmost eye's rays.
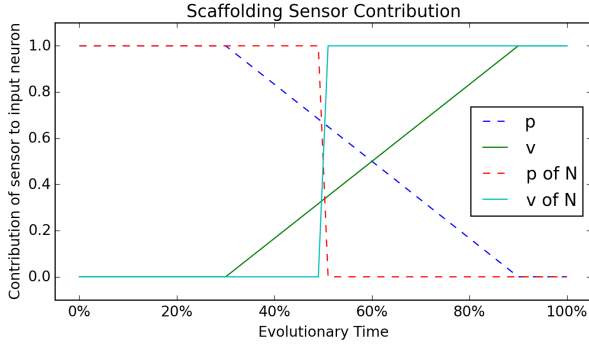

Figure 4: The relative contribution of proprioception (P) and vision (V) to a robot's input over the course of an evolutionary run that is scaffolded. This parameter is then used in each simulation as seen in Figure 5.

During the evolutionary process, the robot is forced to rely progressively less on proprioception and progressively more on vision to perform categorization. Three different types of scaffolds were attempted and reported here. For each scaffolding type, a single parameter linearly descends from one to zero over the course of an evolutionary run and dictates how much proprioceptive input the robot has access to (blue line in Figure 4). A second parameter climbs from zero to one over the course of an evolutionary run and dictates how much visual input the robot has access to (green line in Figure 4). During testing, the controllers evolved using scaffolding were tested identically to the controllers evolved using the Vision (V) sensor modality.

During scaffolded evolutionary runs the robot could rely only on proprioception during the initial 30% of training. The next 60% of training time caused a constant linear decrease in the scaffold. During the final 10% of training, the robot could only rely on vision. Each robot evaluation was provided with a fraction that was zero during the first 10% of training, some value in $[0, 1]$ during the next 60% of training, and one for the last 10% of training. This value was used to tune the three scaffolding schedules described next.

Melding (X) During the evaluation of an individual robot, the values arriving at the sensor neurons were an admixture of the four proprioceptive and the four visual sensors (Fig. 5a). The proportions of both sensor modalities gradually changed over evolutionary time:
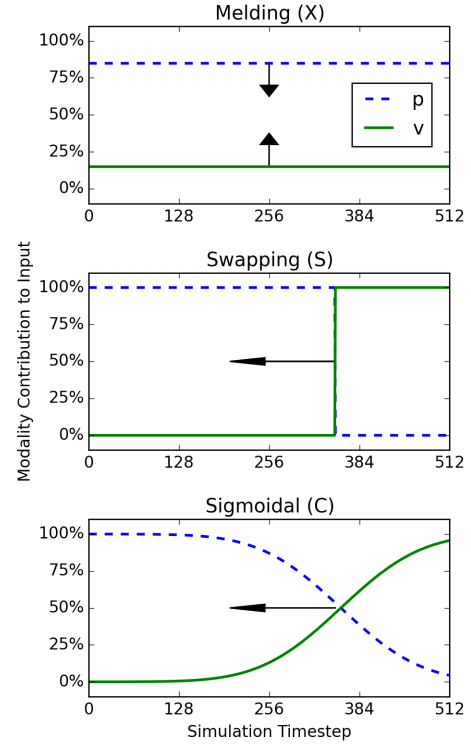

Figure 5: Changes in contribution of proprioception (P) and vision (V) during the evaluation of a single controller. The lines represent ontological scaffolding, or the scaffolding that occurs over one simulation of the robot. The arrows represent how the relative contribution of proprioception and vision change as the evolutionary run proceeds. The movement in the direction of the arrows, as described in Figure 4, represents evolutionary scaffolding.

robots in the first generation obtained 100% proprioceptive input and 0% visual input, robots halfway through an evolutionary run received roughly 50% proprioceptive input and 50% visual input, and robots in the final generation received 100% visual input.

Swapping (S) Partway through the evaluation of a single robot, its input would switch from proprioception to vision (Fig. 5b). The point at which this swap would occur changed over evolutionary time: robots in the first generation received only proprioceptive in-

put, robots halfway through an evolutionary run received proprioceptive input for the first 256 time steps and visual input for the last 256 time steps, and robots in the last generation received only visual input.

Sigmoidal (C) A sigmoidal smoothing function was used to determine the amount of contribution of vision and proprioception to the input layer during any single time step of the evaluation (Fig. 5c). The shape of this sigmoid was altered over the course of an evolutionary run such that the contribution of proprioception dropped more precipitously—and the amount of visual input increased more precipitously—later during the evolutionary run. Essentially, this scaffold is a combination of the other two scaffolds.

None (N) For the first half of evolutionary time the robot's controller solely received input from its proprioceptive sensors. For the second half of evolutionary time the robot's controller solely received input from its vision sensors.

In the case of the sigmoidal smoothing function the contribution of vision to the value of the input neurons is shown in equation 4. The term g represents the current generation in the evolutionary run out of G generations. The term t represents the time step in the simulation out of T time steps.

$$c_v = \frac{\text{erf}\left[4\left(\frac{g}{G} + 2\frac{t}{T} - 1\right) - 2\right] + 1}{2} \quad (4)$$

2.7 Fitness

Each simulation lasted 512 time steps in the Bullet Physics Engine[7]. The final 10% of values of the controller's guess neuron were recorded and used to compute the controller's fitness. The guess neuron's values were compared against the cylinder's class label (-0.5 for small and 0.5 for large) to obtain a difference. This difference is averaged over the time steps to become our error:

$$e = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{T}\sum_{t=0.9T}^{T}|g_{c,t} - r_c| \quad (5)$$

where C represents the number of cylinders placed and T represents the total number of time steps for an evaluation. $g_{c,t}$ denotes the value of the guess neuron when the robot is simulated under environment c and r_c denotes the relative radius of the object in environment c. ($r = -0.5$ for the small object and $r = 0.5$ for the large objects.) In this way an error of zero indicates perfect and stable categorization over the last 5% of the robot's evaluation period. Importantly, the category values were not set to the extrema of the neuron's output range (-1 and +1) because this made the robot's task too easy: instead, it had to hold each guess neuron steady at the correct value for a protracted period of time.

The robot morphology and task were formulated such that there were at least four types of movement that could be used to manipulate objects. The robot could choose to not move objects by extending its joints outward. The robot could open one of its arms while closing the other to slide objects which come into contact with the closing arm away

from it. The robot could close its inner joints while keeping its outer joints relatively open, leading to the object becoming trapped in a diamond-like arm pattern. Finally, the robot could fully close both arms, leading to the object becoming trapped in a triangle formed by the arms. (Figure 1) We found that the controller rarely changed its motion strategy partway through a simulation.

2.8 Tests

After evolution, we assessed how robustly a robot could categorize when simulated in novel environments. To do so, we extracted the controller with the lowest training error obtained during the final 10% of the generations from each evolutionary run. This robot was denoted as that run's representative. The representative controllers were then presented with the Testing environment as shown in Fig. 3. In these test evaluations the robots were only allowed to use the visual sensors for categorization. The only exception were those runs in which only proprioception was allowed during training; these robots were allowed to use only proprioception during testing. As during training, testing error was calculated using Equation (5), but averaged over 156 simulations instead of four, six, or eight simulations.

In the next section we investigate the intra- and inter-category distances between objects caused by the robot's movement. The following equations describe these values. In each case I and J represent the number of large and small objects, respectively.

$$D_{\text{intra}} = \frac{2\sum_{i=1}^{I}\sum_{j=i+1}^{I}\sqrt{(x_i - x_j)^2 + (z_i - z_j)^2}}{I(I-1)} \quad (6)$$
$$+ \frac{2\sum_{i=1}^{J}\sum_{j=i+1}^{J}\sqrt{(x_i - x_j)^2 + (z_i - z_j)^2}}{J(J-1)}$$
$$D_{\text{inter}} = \frac{1}{IJ}\sum_{i=1}^{I}\sum_{j=1}^{J}\sqrt{(x_i - x_j)^2 + (z_i - z_j)^2} \quad (7)$$

3. RESULTS

In this section we report on a total of $3 \times 3 \times 6 \times 50 = 2250$ evolutionary runs. We evolved the robot's controllers against every environment: the combination of object position (horizontal, vertical, and alternating) and simulation count (4, 6, 8). Robots had six modalities: just proprioceptive input (P), just visual input (V), or evolved against one of the four scaffolding strategies (N, S, X, C). For each of the 54 combinations of object positioning, simulation count, and scaffolding strategy, we performed 50 evolutionary runs. For robots trained against four, six, and eight objects, they were evolved for 40,000, 60,000, and 80,000 robot simulations, respectively. This provided every evolutionary run with the same number of evaluations.

The average testing errors for the representative controllers is reported in Table 1. A robot whose strategy would be to randomly guess the size of its cylinders would have a test error of 0.5. When we refer to robots as memorizing we mean that their test error is high; these robots have overfitted the training examples and therefore cannot perform well on the generalized test set.

In most cases, the robots trained with vision (column 'V' in Table 1) memorized more than robots trained using one of the scaffolds (columns 'N' through 'C' in Table 1). How-

	P	V	N	X	S	C
A8	0.16**	0.24	0.23	0.21	0.22	0.19*
H8	0.14***	0.23	0.21	0.21	0.24	0.21
V8	0.16***	0.27	0.25	0.25	0.32	0.27
A6	0.21***	0.31	0.27	0.22***	0.27	0.25**
H6	0.20	0.22	0.25	0.22	0.26*	0.21
V6	0.23***	0.41	0.35**	0.30***	0.34***	0.35***
A4	0.36	0.35	0.34	0.37	0.36	0.39
H4	0.20***	0.36	0.26***	0.26***	0.28***	0.24***
V4	0.20***	0.41	0.34*	0.35*	0.34**	0.36*

Table 1: Test errors for 50 runs over the different object counts. positions, and sensor modalities. The asterisks designate p values below 0.05, 0.01, and 0.001 for one through three asterisks respectively. p values were calculated by applying a t-test to the average test errors of vision when compared to other those of the other sensor modalities for each of the environments.

ever, robots trained with proprioception and then tested using proprioception also memorized on occasion: these robots obtained similarly high testing error as the robots trained and tested with vision in environments H6 and A4. This result implies that although the task may seem sufficiently simple that categorization using proprioception always results in robust categorization in unseen environments, there are movement strategies that evolve for which this is not the case. In this case, the P solutions evolved behaviors that would swing the arms asymmetrically, utilizing feedback from the objects' positions to complete the task of deciding their size.

4. DISCUSSION

It was found that many scaffolded robots evolved to rely on proprioception early during an evolutionary run. These ACP-evolved behaviors contributed to contributed to the evolution of subsequent controllers that exhibited robust visual categorization in novel environments. This is indicated by the significantly lower testing error obtained by many of the scaffolding schedules (X, S, and C) compared to the runs in which only vision was available (V). The behavior exhibited by one of these robustly-categorizing robots is illustrated in Figure 1. As can be seen, this robot's evolved behavior of closing its arms together has the effect of moving objects at different positions to the same position directly in front of the robot. This has the result of reducing differences in irrelevant properties of the object; here, such an irrelevant difference is the different positions of the objects.

In contrast, a robot that does not move will generate no difference in sensor signatures during different object placements if it relies on proprioception for categorization, and very different sensor signatures if it relies on vision. Neither bode well for robust categorization in unseen environments. In the former case, the robot will not be able to successfully categorize even under training environments. In the latter case, there is a danger that the robot will memorize the training environments and fail to generalize to any unseen environments. This highlights the importance of motion for active categorical perception and that proprioception is more likely to lead to active behaviors: a blind robot must move and contact objects in order to categorize them.

4.1 Scaffolding success through motion

For the experiment set involving vertical arrangement of six object positions (V6), we obtained some of our most successful results. Since the training set consisted of closely positioned objects, vision-evolved controllers had a natural tendency to memorize with little movement. As shown in Figure 6 both the proprioception and all of the scaffolded runs resulted in significantly more motion during testing. This indicates that when vision favors passive behaviors that do not involve object manipulation, then scaffolding can be a good way to bias search toward movement-based categorization. This movement-bias is retained while the robot transitions to vision, and results in increased robustness of the eventual visual classifier.

One of the primary indicators for whether a controller would generalize was the extent to which it manipulated the object. As shown in Figure 6, the motion induced by the vision-based controllers is significantly lower than any observed in the scaffolded runs. Memorization combined with the lack of motion is the reason that the visual classifier was only able to successfully categorize objects inside the range of its training positions.

The scaffolding process can therefore lead to robust visual classifiers. The efficacy of scaffolding indeed increased as the training set grew increasingly sparse (eight objects are reduced to six and then four in Table 1) and accordingly the amount of computational effort available was increasingly restricted (from 80,000 robot simulations to 60,000 to 40,000).

4.2 Scaffolding success in other cases

We also investigated the effect of scaffolding when the visual classifier's motion was not significantly different from robots that relied on proprioception. This was the case for the A8 training regimen, as shown in Figure 7. However, even in this case, the C scaffolding schedule achieved significantly lower test error than pure vision.

The reason for this is that motion is not a meaningful metric in and of itself. A robot may evolve to move its arms a great deal and push the objects away from it in ways that exaggerate the irrelevant feature of object position.

To distinguish between helpful and unhelpful motion, we can look at intra-category and inter-category distances. Intra-category distance, the average distance between the final position of an object and every other object in its category, would be low for the behavior shown in Figure 1 as the objects would be pulled to about the same location. Since objects are getting pulled close regardless of size, we would expect to see inter-category distance, the average distance between an object and every other object not in its category, to also decrease a similar amount.

Because the radii of the objects are different, we do not expect inter-category distances to be lower than intra-category distances as the centers of the two object sizes would be in marginally different places (25% of the small object's radius) when the objects are flush against the robot's chassis. For unhelpful movement, objects may be pushed away from a swinging arm or not moved at all: both intra-category and inter-category distances should thus remain high. The results in Figure 7 show that the scaffolds that were most successful have intra-category and inter-category differences that are low, like those for proprioception. The unsuccessful scaffold (S) characteristically evolved higher intra-category
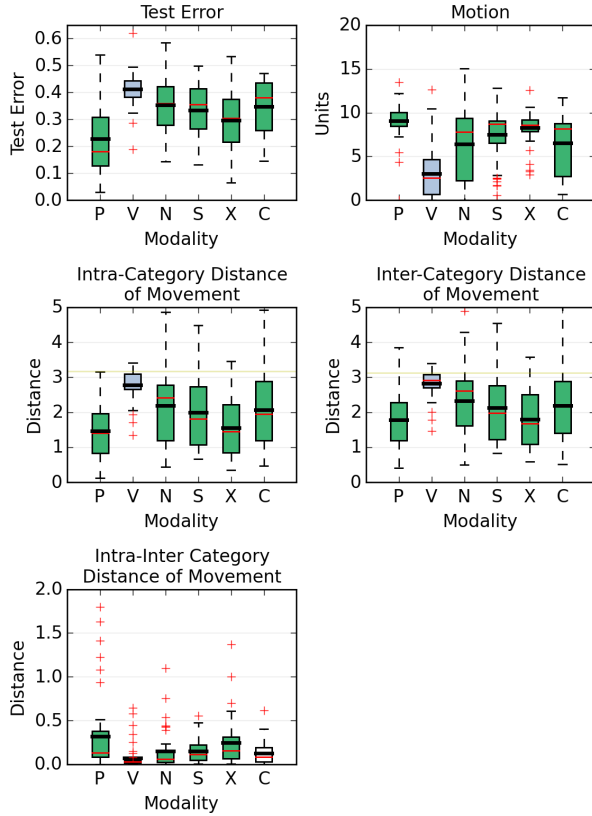
Figure 6: Statistics of V6-trained controllers over 60,000 simulations per run. We define motion as the average euclidean distance between the beginning and ending positions of the objects during testing simulations. The light blue boxplot represents vision. Green boxplots for each subplot are significantly different than vision at a p level of 0.05. The horizontal red lines designate medians and the thick horizontal black lines designate the mean. In the intra and inter-category graphs the horizontal yellow lines designate what the distances would be if the test objects were not perturbed. The boxplot's whiskers represent the 25th and 75th percentiles.

and inter-category behavior, which were more in line with the same metrics for the pure vision runs (V).

From this, it seems likely that the best predictor of whether a particular run will produce robust visual classifiers is whether the difference between intra-category and inter-category distances is magnified by motion induced by the robot's limbs. Indeed this is what is observed in the results from the V6 training regimen (Figure 6).

The types of movements that the scaffolds can help instigate is therefore also an important component of whether they lead to robust visual categorization, and the signature of whether motion is helpful is if it reduces the separation between intra-category and inter-category differences.

4.3 Scaffolding Issues

As shown in Table 1, both the vertical and horizontal environments' scaffolds lead to relatively better generalizers as we provide fewer training positions, and therefore less computational power. This highlights vision's inclination towards memorization. In the case of the alternating ob-
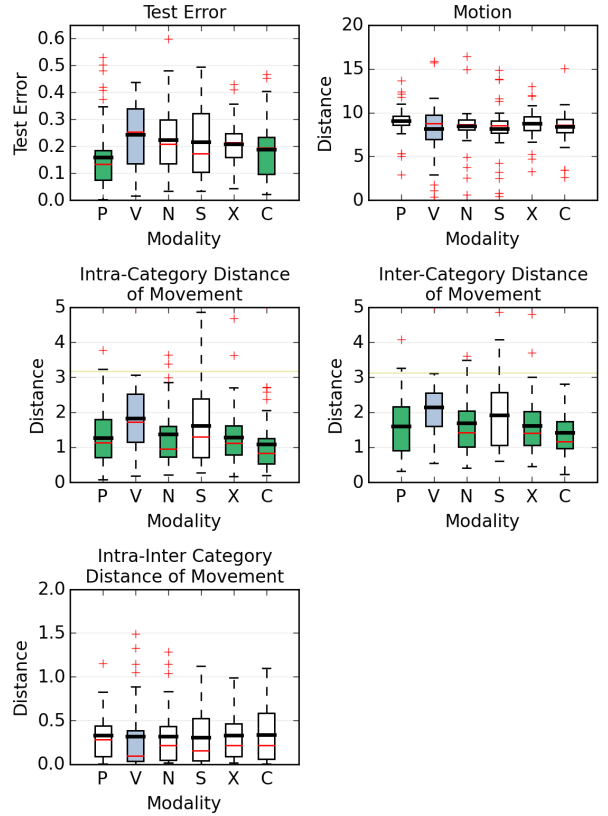

Figure 7: Statistics of A8-trained controllers over 80,000 simulations per run.

ject positions, a different pattern emerges. In the case of A4, neither proprioception nor any of the scaffolds have significantly different means; proprioception becomes just as much a memorizer as vision. This explains the lack of success of the scaffolds; they do not have a robust categorization strategy from which to begin weaning the robot off proprioception. However, as we add computational power and complexity through A6 and A8, proprioception-based robots memorize less. Even as the environments exhibit greater variation and vision-based controllers memorize, proprioception based controllers resist memorization and are thus still able to be scaffolded. This indicates that even when problems are not constrained to a single dimension of position, there may be success through sensor scaffolding. The underlying pattern for success is whether proprioception can evolve and then pass these successful grasping behaviors to vision. The grasping behaviors that work are the ones that collapse the state space by reducing intra-category and inter-category distances.

When comparing our four scaffolds, none of the scaffolds had a clear and universal advantage over any other. The positive aspect of this is that scaffolding strategies are a manual process that the experiment designer must consider. On the other hand, we still have little insight into the underlying intricacies of applying different scheduling scaffolds.

5. CONCLUSIONS AND FUTURE WORK

Here we have demonstrated how, through action, a robot may be gradually transitioned from active categorical perception to visual classification.

Direct successors to this research revolve around improving the efficacy of scaffolding. We believe that the current method presented in this paper can potentially benefit from further optimizations. These potential investigations include spending more time evolving proprioceptive behaviors and limiting evolution's ability to move away from ACP-evolved grasping behaviors. Another angle of approach would be provide each evolutionary run with a randomized set of initial positions to evolve with. Other future work might include evolving classifiers that utilize both touch and vision concurrently, or a system that learns to map sensors to neural inputs concurrently with the scaling of its neural weights.

The results presented here and previously in [3], when taken together, suggest a pathway for uniting the two sister disciplines of evolutionary robotics and developmental robotics. Work in evolutionary robotics tends to focus on how control and morphology can be gradually shaped by evolutionary pressures to enable successful achievement of a task [6]. In developmental robotics, the focus is often on how an individual robot gradually acquires greater behavioral competency as its sensory and motor systems complexify or become less constrained [15].

If these two approaches were combined in future work, evolution may explore different kinds of developmental transitions from embodied to non-embodied categorization, beyond the four manually-devised transitions we studied here. Evolved transitions that enable a more efficient transition from proprioception to vision during the robot's lifetime would confer an evolutionary advantage on it, thus leading to the evolution of increasingly efficient transitions. This better reflects biological evolution, which evolves developmental trajectories from infant to adult forms, rather than fixed traits. Such a combined evolutionary and developmental approach may discover evolved transitions that are more efficient and effective than the manually-devised developmental transition studied here.

Furthermore, in previous work [3] it was shown that evolving morphology can facilitate the acquisition of active categorical perception in robots. Thus, evolving morphology may further empower evolution to discover useful control, morphology and action combinations that lead to efficient transitions from embodied to non-embodied categorization.

It may also be applicable to other aspects of cognition. For example, it may be possible to automatically evolve embodied language understanding [9] or embodied symbol manipulation [14], and then automatically and gradually transition these competencies to abstract reasoning about language or symbols.

6. REFERENCES

[1] R. D. Beer. The dynamics of active categorical perception in an evolved model agent. *Adaptive Behavior*, 11(4):209–243, 2003.

[2] Y. Bengio. Learning deep architectures for AI. *Foundations and Trends® in Machine Learning*, 2(1):1–127, 2009.

[3] J. Bongard. The utility of evolving simulated robot morphology increases with task complexity for object manipulation. *Artificial Life*, 16(3):201–223, 2010.

[4] J. Bongard. Morphological and environmental scaffolding synergize when evolving robot controllers: artificial life/robotics/evolvable hardware. In *Proceedings of the 13th annual conference on Genetic and Evolutionary Computation*, pages 179–186. ACM, 2011.

[5] J. Bongard. Morphological change in machines accelerates the evolution of robust behavior. *Proceedings of the National Academy of Sciences*, 108(4):1234–1239, 2011.

[6] J. C. Bongard. Evolutionary robotics. *Communications of the ACM*, 56(8):74–83, 2013.

[7] E. Coumans et al. Bullet Physics Library, version 2.82. http://bulletphysics.org, 2014.

[8] M. Dorigo and M. Colombetti. Robot Shaping: Developing Autonomous Agents Through Learning. *Artificial Intelligence*, 71(2):321–370, 1994.

[9] M. H. Fischer and R. A. Zwaan. Embodied language: A review of the role of the motor system in language comprehension. *The Quarterly Journal of Experimental Psychology*, 61(6):825–850, 2008.

[10] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini. Learning about objects through action: initial steps towards artificial cognition. In *IEEE Intl Conf on Robotics and Automation*, volume 3, pages 3140–3145. IEEE, 2003.

[11] N. Hansen. Cma evolution strategy, 1.1.05. https://www.lri.fr/ hansen/cmaes_inmatlab.html, 2014.

[12] S. Harnad. To cognize is to categorize: Cognition is Categorization. *Handbook of Categorization in Cognitive Science*, pages 20–45, 2005.

[13] G. E. Hinton. Learning multiple layers of representation. *Trends in Cognitive Sciences*, 11(10):428–434, 2007.

[14] G. Lakoff and R. E. Núñez. *Where mathematics comes from: How the embodied mind brings mathematics into being.* Basic Books, 2000.

[15] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini. Developmental robotics: a survey. *Connection Science*, 15(4):151–190, 2003.

[16] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *IEEE CVPR*, 2015.

[17] S. Perkins and G. Hayes. *Robot shaping: Principles, methods and architectures.* 1996.

[18] J. Plumert and P. Nichols-Whitehead. Parental scaffolding of young children. *Developmental Psychology*, 32(3):523–32, 1996.

[19] L. M. Saksida, S. M. Raymond, and D. S. Touretzky. Shaping robot behavior using principles from instrumental conditioning. *Robotics and Autonomous Systems*, 22(3):231–249, 1997.

[20] C. Scheier and R. Pfeifer. Classification as sensory-motor coordination. In *Advances in Artificial Life*, pages 657–667. Springer, 1995.

[21] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *International Conference on Learning Representations*, 2014.

[22] E. Tuci, G. Massera, and S. Nolfi. Active categorical perception of object shapes in a simulated anthropomorphic robotic arm. *IEEE Transactions on Evolutionary Computation*, 14(6):885–899, 2010.