Autonomous Construction and Use of Simple to Complex Tools

A Introduction

Goal: The goal of this project is to develop methods for autonomous tool construction and use in sensorimotor agents.

Background and Gap: Intelligence is a multi-faceted capability, spanning various skills and requiring diverse knowledge (Shumaker et al. 2011). Would it be possible to come up with a single task that is (1) simple and concrete yet (2) covers multiple skills, and is (3) extensible within its original scope? Common indicators of intelligence include communication (Snowdon 1990), logic (Allen 2006), and tool construction/use (Hall 1963). Communication (language) and logical inference have been extensively investigated in artificial intelligence (AI). However, tool construction and use is still a largely under-developed area in AI, especially when it comes to tool construction.

Objectives: This project will design an environment where sensorimotor controllers can be adapted to construct and use tools of greater complexity and utility based on minimal task knowledge (Ambrose 2001). A step-by-step approach will be taken to facilitate such an autonomous use of tools: (1) Using simple tools for reaching. (2) Constructing composite tools for reaching. (3) Constructing composite tools for dragging. (4) Tool construction and use in real-world physics simulation. (5) Analysis of sensorimotor controllers optimized for each task.

Approach: A combination of neural networks, genetic optimization, modular learning, and unsupervised learning will be used for the tool construction and use tasks. The Box2D platform will be used for realistic physics simulations.

Intellectual merit: Construction and use of complex tools is one of the rare intelligence-related skills unique to humans. Although simple tool construction/use has been observed in animals, the complexity of those tools is far below that of the human. Investigating how the capability to construct and use tools can emerge in autonomous sensorimotor agents can lead to greater insights on the nature of intelligence. Analysis of the resulting sensorimotor controllers can help us better understand the co-adaptive aspect of intelligence and tools.

Broader impacts: One graduate student and one undergraduate student (through the Research Experience for Undergraduates supplement) will be trained as part of this project. An open simulation environment will be developed for tool construction and use, and the platform will be used to initiate a tool construction and use challenge targeted at the AI, machine learning, and robotics communities. The simulation platform will also be used for K-12 and public outreach. All software and data will be published and then released as an open-source project (GNU Public License or similar). The project is expected to have broader impacts on robotics, cognitive science, ethology, and anthropology.

B Background

B.1 Tool Construction and Use in Animals

Tool construction and use require high levels of sensorimotor skill, planning, and problem solving capabilities. Only a small number of animals exhibit the capability of tool use and yet fewer are known to construct tools (see Shumaker et al. 2011 for an overview).

Tool use is observed in various animals, not just in humans. For example, chimpanzees use primitive tools such as stones or sticks (Boesch and Boesch 1990; Whiten et al. 1999), macaque monkeys use stone axes for various purposes (Gumert et al. 2009), parrots use sticks to reach objects beyond a barrier (Auersperg et al. 2014), Degus (a South American rodent species) can be trained to use a rake-like tool to obtain food under a fence (Okanoya et al. 2008), elephants also use sticks or other objects to obtain food located out of normal reach (Foerder et al. 2011), and dolphins use sponges to cover their snout when digging for food (Smolker et al. 1997).

Tool construction is only rarely observed in non-human animals. Chimpanzees have long been known to construct simple tools, such as a stack of boxes to reach high-hanging fruit (Kohler 1924). More recently, Price et al. (2009) showed that Chimps, after training, can put together two sticks to extend the reach of the tool. Non-primate species have also shown tool construction capability, although limited in its complexity. For example, New Caledonian crows in captivity have been observed to bend a wire to create a hook to retrieve food (Kenward et al. 2005). However, tool construction (especially in the wild) is extremely rare among non-human animals.

Fig. 1 below shows levels of tool construction and use with increasing difficulty (Choe et al. 2015). Level 1 corresponds to the use of unmodified objects as a tool, which is most commonly observed in animals. Level 2 represents the construction of simple tools through modification of an object or putting together a small number of objects. Level 3 is where multiple objects are put together to form a more complex tool. Level 4 involves two or more agents cooperating to construct a tool, including abstract (or social) tools. Level 5 calls for the ability to explain how tools are constructed, used, and why. Animals other than humans have only reached level 2.

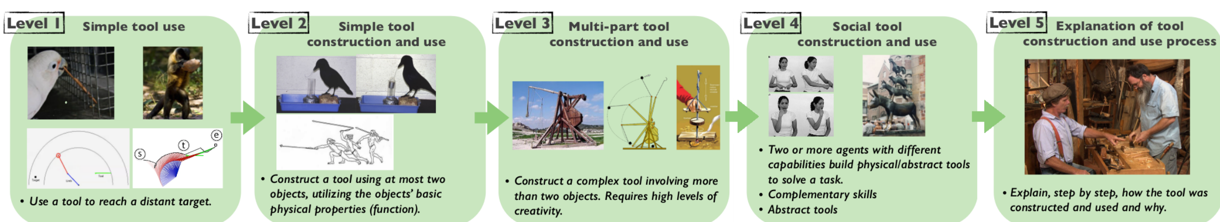


Figure 1: **Levels of Tool Construction and Usage.** Based on the PI's white paper (Choe et al. 2015), proposing the use of tool construction and use as a measure of intelligence (cf. St. Amant and Wood 2005). Image sources: Level 1: Auersperg et al. (2014), BBC, Li et al. (2015). Level 2: Kenward et al. (2005), Cowley (1996). Level 3: ChrisO (2005); Onno (2008); Ruiz (2014). Level 4: Photos by Ann Senghas (Wikimedia commons) and Adrian Pingstone (public domain). Level 5: PBS. Note that non-human animals are limited to levels 1 and 2. Existing AI and robotics works are limited to level 1. In this project, we will focus on levels 1 through 2.

B.2 Tool Construction and Use in AI and Robotics

Tool use has recently gained attention in artificial intelligence and robotics (for a review, see St. Amant and Wood 2005), however tool construction is a relatively unexplored area. The various existing works include (1) programmed, hard-coded behavior (Murphy 2000); (2) learning through demonstration (Arsenio 2004; Lee et al. 2008; Pastor et al. 2009; Saegusa et al. 2014; Wu and Demiris 2011); (3) learning affordances via random trial-and-error or body babbling (Bullock et al. 1993; Katz and Brock 2008; Stoytchev 2005); (4) tool use based on tool-body assimilation (Nishide et al. 2012; Takahshi et al. 2014,b). (5) bayesian learning of tool affordances (Jain and Inamura 2014); and (6) Evolved tool using behavior (Chung and Choe 2011a; Schäfer et al. 2007). (Cf. Sims 1994 where body morphology [not tools] was co-evolved with the controller.)

However, most of the works listed above depended on some degree of designer knowledge regarding tool use and motor control, e.g., fully hard coded behavior, the tool being pre-attached to the limb, pre-defined tool features, pre-defined motor primitives, etc. Evolution-based approaches (Chung and Choe 2011a; Schäfer et al. 2007) were relatively free of these constraints, but in those cases the tools were more or less simple markers, not something than can be manipulated with a limb-like structure of the agent.

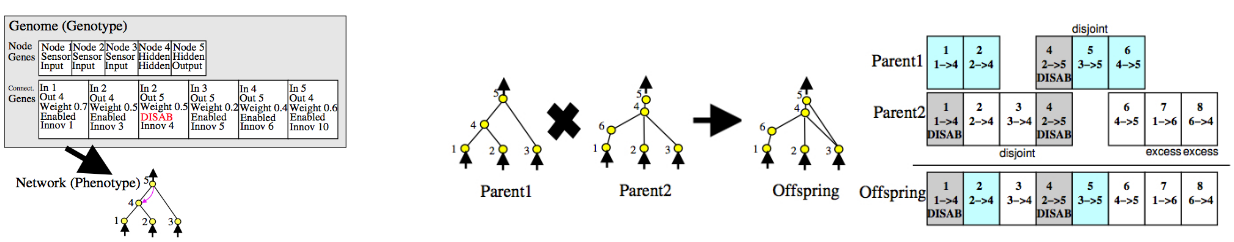
AI-based work on tool construction is almost non-existent. Wang et al. (2014) took a synthetic approach to construct tools, but their focus was more on understanding the various mechanistic and energetic needs of using the synthetically generated tools, not on tool construction by agents.

B.3 Synthetic Evolution of Neural Circuits

Early efforts in neuroevolution were limited to adjusting the connection weight, while leaving the neural circuit topology fixed (Montana and Davis 1989; Whitley et al. 1993; Wieland 1990). In these approaches, each genotype was mapped to a full neural network. However, these approaches were not flexible enough and could not handle increasing levels of task complexity. More recent approaches were based on single neuron-level evolution (Agogino et al. 2005; Gomez and Miikkulainen 1997; Moriarty and Miikkulainen 1997; Potter and De Jong 2000), however, the evolved neurons had to be assembled into a network, which typically had a fixed topology.

In this project, we need a method that allows the network topology to evolve, thus the methods above are not suitable. There are several approaches that allow network topology evolution (Fullmer and Miikkulainen 1992; Yao 1999), however, these were based on weight-topology co-optimization, thus, they were not flexible enough. For our project, we will use a neuroevolution technique called the Neuroevolution of Augmenting Topologies, or NEAT (Stanley and Miikkulainen 2002). Unlike most other neuroevolution techniques, NEAT allows the neural circuit to evolve to have an arbitrary connection topology.

In NEAT, the chromosome encodes neurons and their connections separately, as well as the connection weights. Neurons and connections can be added or removed to change the network topology, thus the chromosome has a variable length. Mating of chromosomes with different network topology is achieved through the use of a quantity called "innovation number", unique to each gene, that indicates the evolutionary origin of that particular gene. Innovation numbers allow only compatible genes to mate (i.e., genes that have the same ancestral origin).



(a) Genotype (b) Crossover

Figure 2: **Neuroevolution of Augmenting Topologies (NEAT).** (a) The genotype-to-phenotype mapping in NEAT is shown. Each node and each connection has a gene. Each connection gene has an enable/disable flag and a unique identifier, the *innovation number*. (b) Crossover of two parents with different topology is shown. The genes of the two parents are aligned so that the innovation numbers match up. Adapted from Stanley and Miikkulainen (2002).

See Fig. 2 above for the genotype to phenotype mapping, and crossover of topologically different networks. Mutation is not shown in the figure but its implementation is straight-forward (insert or delete connections or neurons). Another unique mechanism of NEAT is “speciation” that helps freshly changed topology to be preserved despite the initial plunge in fitness. The rest of the algorithm is similar to other neuroevolution or evolutionary algorithms: instantiate phenotype from genotype $\rightarrow$ test in the task environment $\rightarrow$ calculate fitness $\rightarrow$ selection and reproduction.

C Prior Work

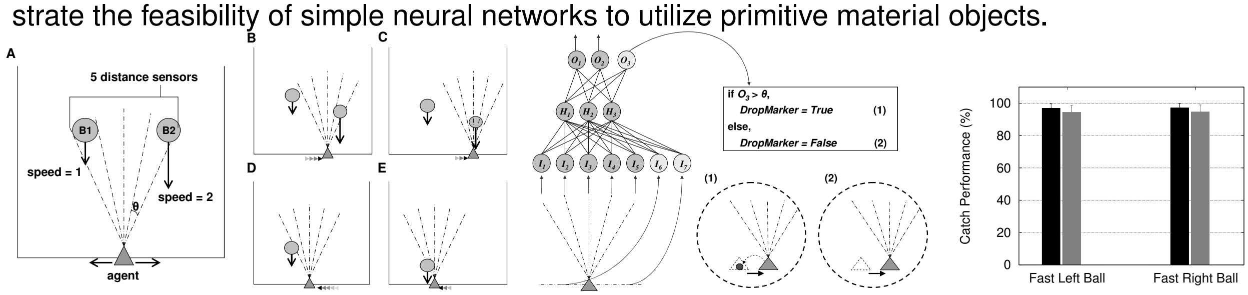
C.1 Using Simple Environmental Markers as a Tool

Can simple evolved neural networks utilize tools? Our prior work on evolved neural controllers in simple sensorimotor tasks such as ball catching (Chung and Choe 2009) and foraging (Chung and Choe 2011b) demonstrates that external objects can be used as markers to serve as a form of external memory.

We tested whether evolved feedforward neural networks can exhibit memory related behavior, if the ability to drop and detect external environmental markers is allowed. Feedforward neural networks are reactive, thus their output only depends on the immediate input, not past input, thus they cannot solve such problems. For example, the ball catching task (Fig. 3a) or the foraging task (Fig. 4a) requires memory: catch the first ball and return to the second ball that went out of sensor range (Fig. 3a); or get food from the first target within the sensor range, bring it back to the nest, and then go back to get the food from the second target that was seen earlier while at the first target (Fig. 4a). The agent was allowed to drop and detect markers in the environment (Fig. 3b and 4c).

Although the evolved neural circuits are feedforward networks, with the addition of the external marker dropper/detector, we have been able to evolve agents that can exhibit memory behavior (Fig. 3c and 4d). The performance was comparable to networks with built-in dynamic memory (recurrent networks) without the dropper/detector. Interestingly, behavioral patterns of dropper/detector networks also showed more economy (Fig. 4c; recurrent network data not shown).

These results show the importance of taking into consideration environmental factors and how the agent can affect them through sensorimotor interaction. Furthermore, these results demonstrate the feasibility of simple neural networks to utilize primitive material objects.

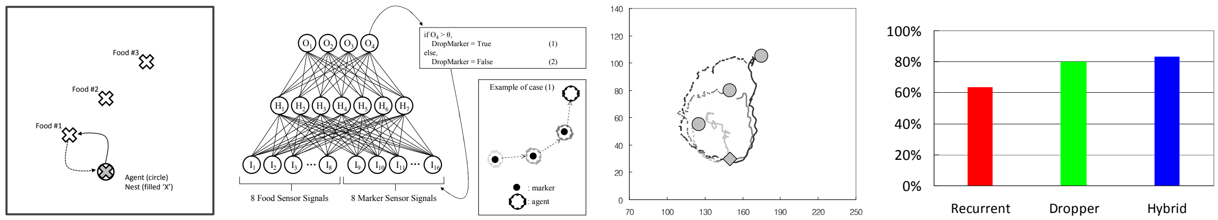


(a) Task (b) Dropper/detector network (c) Performance

Figure 3: **Dropper/detector network’s performance in ball catching.** (a) Agent needs to catch both balls (A). Moving toward the fast ball (B) makes the slow ball out of sensor range (C). To come back for the slow ball (D-E), memory is needed. (b) The dropper/detector network (feedforward). (c) Performance of dropper/detector network (gray) vs. a recurrent network (black). Feedforward networks max out at 50% (only one ball caught). Adapted from PI’s work (Chung et al. 2009).

C.2 Self-Organization of Target-Reaching Gesture Maps

Before an agent can learn to use a tool, it needs to understand its own body, a map of its own behavior. The motor cortex in the macaque brain forms a topographical map of complex behaviors

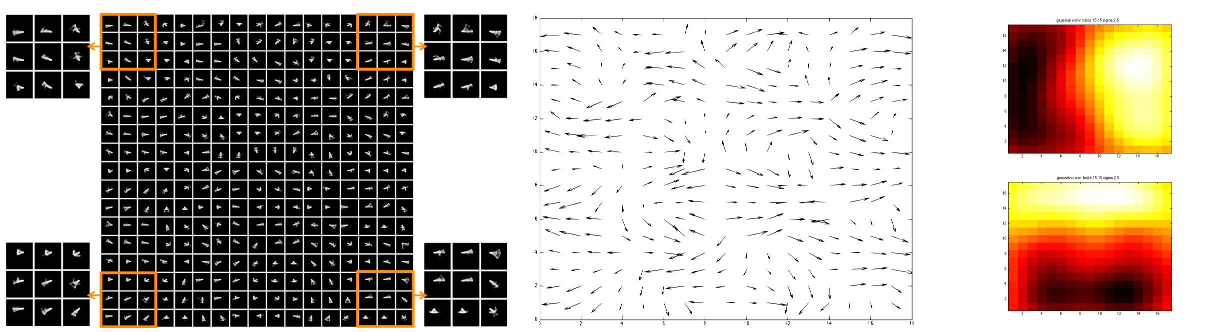


(a) Foraging task (b) Controller (c) Behavior (d) Performance

Figure 4: **2D foraging task and dropper/detector network.** (a) 2D foraging task: The x mark at the bottom is the nest, and the top three are food sources. The agent, shown as a circle, is at the nest. The sensor range is shown in red. (b) The controller network for the foraging agent is shown. The network is again a feedforward network, with additional sensors for marker detection and an added output for marker dropping. (c) Sample trajectory of a dropper/detector network. (d) Performance comparison is shown for three types of networks, recurrent network (red), dropper/detector network (green), and hybrid (recurrent network with dropper/detector, blue). Feedforward networks cannot solve this task (data not shown). Adapted from PI's work (Chung and Choe 2011b).

(Graziano et al. 2002), where the final posture of the movements form a topologically organized map. For example, one part of the motor cortex maps the target location in reaching behavior, where the map shows a topological organization (nearby neurons represent nearby target locations in the environment).

In our prior work, we used a computational model of cortical development called GCAL (Gain Control, Adaptation, Laterally connected) model (Bednar 2012; Law et al. 2011), a simplified yet enhanced version of the LISSOM model by the PI and his colleagues (Miikkulainen et al. 2005). GCAL is basically an unsupervised learning algorithm, mimicking synaptic plasticity in the cortex. We converted an articulated arm with two joints and its random reaching behavior (body babbling; Chung et al. 2015) into time-lapse images, and used these to train the GCAL model. We used the Topographica neural map simulator (the PI contributed partly) to self-organize target-reaching gesture maps in a computational model of the developing cortex. Preliminary results are shown in Fig. 5. The results show that a topological map similar to those observed in the macaque motor cortex can be obtained. These results will help us form the body-model of our tool using agent.



(a) Gesture map (b) Target direction map (c) Horiz./Vert. component

Figure 5: **Self-Organized Target-Reaching Gesture Map.** Self-organized target-reaching gesture map. (a) A plot of an 18×18 map of reaching gestures of a two-joint arm. Each grid shows a time-lapse image of a reaching gesture. (b) Target direction map. The arrow heads point to the final position of the end effector relative to the first joint. (c) Top: horizontal components of the target direction map; Bottom: vertical components of the target direction map (dark: $\leftarrow$,$\downarrow$, bright: $\rightarrow$, $\uparrow$), showing a clear topological organization. Adapted from PI's recent work (Yoo et al. 2014).

C.3 Results from Prior NSF Support

The PI has done an extensive range of research directly relevant to this project (see the biosketch), but has not been able to secure NSF funding for those works, although there were several close calls (Competitive, Fund if possible, etc.). The PI has been supported however on his brain imaging and neuroinformatics work by the NSF, and these results will be discussed below. (a) Award number: #0905041, Amount: $114,024, Period: 09/01/2009–08/31/2012. (b) Title: CRCNS data sharing: Whole Mouse Brain Neuronal Morphology and Neurovasculature Browser. (c) Summary: Intellectual merit: The grant resulted in innovative brain imaging and neuroinformatics frameworks, and an open online mouse brain atlas (Knife-Edge Scanning Microscope Brain Atlas [KESMBA]: Fig. 15c-d). KESMBA included a novel distance-attenuation technique for in-browser 3D visualization, and the OpenLayers API for multiscale navigation. Furthermore, interoperability with other brain atlases (such as the Allen Brain Atlas) has been incorporated. Finally, an interactive data volume viewer using WebGL was developed for free viewing of regions of interest. Broader impacts: We launched the web atlas to deliver our submicrometer-resolution mouse brain atlases ($\sim$ 4.5 TB) and published associated code on SourceForge (project: kesmba). We also ran exhibits at conferences (2012, 2013, 2014, and 2015), organized a tutorial (2013, Fig. 15a), and provided data for a science exhibit at San Francisco's *Exploratorium* (Fig. 15b). We also participated in the national *Discover Engineering* event. Three Ph.D. and ten M.S. and two REU students were trained. (d) Publications: The project (and related NSF projects) resulted in three journal papers (Choe et al. 2011b; Chung et al. 2011; Mayerich et al. 2011b), nine conference papers (Choe et al. 2011a; Kwon et al. 2011; Lal Das et al. 2015; Mayerich et al. 2011a; Sung et al. 2013; Yang and Choe 2010, 2011a,b; Zhang et al. 2015), three Ph.D. dissertations (Chung 2011; Sung 2013; Yang 2011), and ten M.S. theses (Choi 2013; Dileepkumar 2014; Kim 2011; Lal Das 2014; Miller 2014; Shah 2014; Singhal 2015; Srivastava 2015; Yang 2014; Zhang 2014) (e) Data and code are available from our project web site and SourceForge (kesmba).

D Research Plan

Roadmap: A roadmap of the research plan is shown below.

	D.1	D.2	D.3	D.4
Tool	Single (stick)	Composite (extended stick)	Composite (T-shaped)	All
Task	Reaching	Reaching	Dragging	Both
Algorithm	NEAT	NEAT	NEAT+More	NEAT+More
Affordances	Ⓐ	Ⓐ	Ⓐ	Ⓜ
Tool-Body Assimiliation	Ⓐ	Ⓐ	Ⓐ	Ⓜ
	D.5 Analysis			

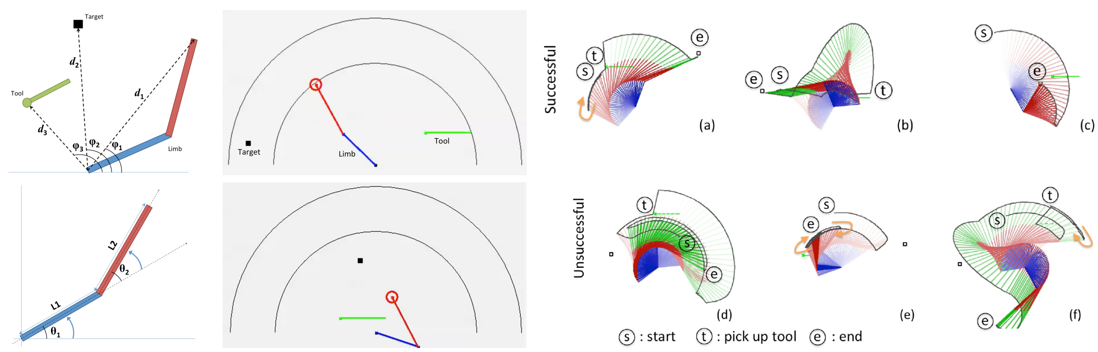
NEAT+More: added mechanisms for modular learning, predictive learning, and auxillary fitness inspired by options and reward-shaping (in reinforcement learning); Ⓐ: assumed, Ⓜ: modeled.

D.1 Using Simple Tools for Reaching

For simple tool use, we will use the NEAT algorithm, unmodified (Section B.3). Some simplifying assumptions will be made, e.g., tool automatically attaching to the end effector once the tool handle is reached, etc. First, we will set up a simulation environment as shown in Fig. 6a. The task is a simple target reaching task, where targets either within or outside of the arm's reach has to be touched (Fig. 6b). For out-of-reach targets, a stick tool needs to be used.

For the sensory representation, we will use a body-centered coordinate system, based on our earlier experience in a simple reaching task (Mann and Choe 2010). The state of the task environment is shown in Fig. 6a, with all the angles and distances involved. The inputs of the controller neural network are computed from those values (Fig. 7a): $HandToTarget = (\varphi_2 - \varphi_1, d_2 - d_1), HandToTool = (\varphi_3 - \varphi_1, d_3 - d_1)$. Additional inputs include the joint angles (θ_1, θ_2) and the joint limit detectors: $\vartheta_{\{\theta_1,\theta_2\}} = 1$ if $\{\theta_1, \theta_2\} \geq 150^\circ$ or $\{\theta_1, \theta_2\} \leq -150^\circ$ and 0 otherwise.

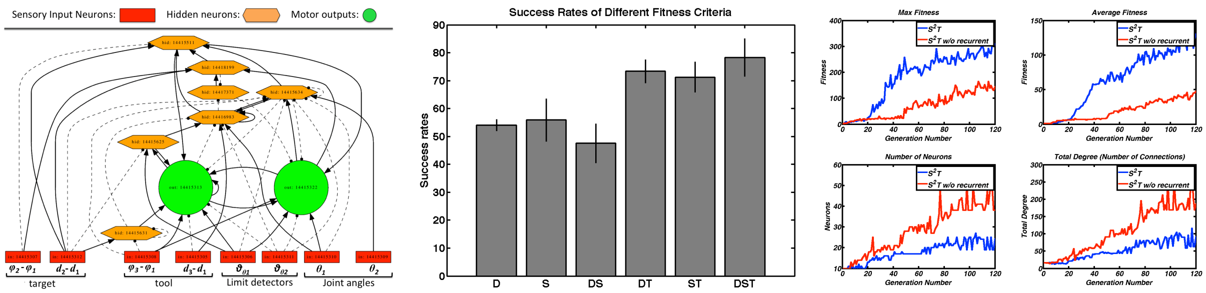
We will use three basic fitness criteria and test different combinations of them to evolve the neural circuit controller for the limb. For each controller, by the end of 100 trials, the following quantities will be calculated: (1) D: distance between the end effector and the target; (2) S: steps taken to reach the target; and (3) T: tool pick-up frequency, defined as $D = 1 - \frac{\sum_k \|\vec{o_k} - \vec{e_k}\|}{KD_{max}}$, $S = 1 - \frac{\sum_k s_k}{KS_{max}}$, $T = \frac{\sum_k t_k}{K}, and t_k =$ 1 if tool is picked up an 0 otherwise, where $\vec{o_k}$ and $\vec{e_k}$ are the coordinates of the target object and the end effector of the limb, respectively. The Euclidean distance $\|\cdot\|$ is normalized by the maximum radius of the environment D_{max}. K indicates the total number of the trials ($K = 100$ in the experiment) and k indicates the k-th trial. s_k indicates the number of steps taken before reaching the target, and S_{max} is the maximum movement steps for each trial ($S_{max} = 500$ in the experiment). Different combinations of the fitness criteria elements will be tested: D, S, DS, DT, ST, and DST. Multiplication will be used when combining the fitness criteria elements ($DS = D \times S$). In addition, neural circuits evolved with and without recurrent connections will be compared ($SST = S^2T$ vs. $SST_{noRecur} = S^2T_{noRecur}$). Preliminary results are shown in Fig. 6c (behavior observed in successful and unsuccessful trials) and Fig. 7b-c (performance using different fitness and learning trend for recurrent vs. feedforward circuits).



(a) Sensors (top) and kinematics (bottom) (b) Out of reach (top) and within reach targets (bottom) (c) Behavioral patterns

Figure 6: **Tool Pick-Up and Reaching Task.** Picking up a tool (or not) and reaching a target using a jointed limb. (a) The sensors (limb angle, target [black box] and stick's [green] angle and distance) and kinematics are shown. (b) Top: target (black box) is beyond the arm's reach. Bottom: target is within the arm's reach. (c) Various evolved reaching behavior (time-lapsed: dark = close to present, light = farther into the past). Target (box) appears in random locations, and limb configuration is also random in the beginning. The agent may or may not pick up the tool. Events are marked (s)tart, (t)ool-pickup, and (e)nd. End-effector trajectories are shown as gray curves. Top: successful trials. Bottom: failed trials. Preliminary result (Li et al. 2015).

Research Issues: As briefly mentioned at the beginning of this section, certain simplifying assumptions are made: (1) the end effector attaches to the tool automatically, effectively extending the last limb segment. (2) A side effect of (1) is that the target angle and distance is now calculated relative to the end of the extended limb. Another implicit assumption is (3) target and tool have separate input representations, thus there is no need for the agent to distinguish the two objects using a shared sensor. Finally, (4) the relative distance and angle of the end effector and the target or tool are precomputed and fed into the controller network. These assumptions make the tool manipulation and interface greatly simplified.



(a) Typical Evolved Controller (b) Performance (c) Learning

Figure 7: **Reaching Task Performance and Learning Curve.** (a) A typical controller network. Inputs (red rectangles) are as follows: target angle, target distance, tool angle, tool distance, joint 1 and 2 angles, and joint 1 and 2 limit detectors. The two outputs (green circles) control joint 1 and 2 angles. Hidden units (orange hexagons) are evolved. (b) Performance (mean and std) based on different fitness criteria: D(istance), S(peed), and T(ool pick up frequency). The results are robust. (c) Learning curve based on different network constraints (with [blue] or without recurrent loops [red]). Learning is faster (top plots) and fewer neurons are used (bottom plots) with recurrent loops. Preliminary results in Li et al. (2015).

Can NEAT learn to use the tool when the task environment is set up so that these assumptions are relaxed? For example, instead of two sets of sensors for target and tool, respectively, what if a single set is used for detecting the closest (or attended) object? We will experiment with relaxed assumptions using an unmodified NEAT implementation to determine feasibility. We expect that part of these issues will have to be addressed using more elaborate methods to be developed for the subsequent tasks in Sec. D.3 and beyond (also see Fig. 11).

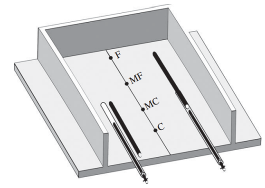
Target reaching performance alone may not be sufficient to evaluate the outcomes. The experiments have to be carefully designed so that default or random behavior does not lead to high performance. For example, target placement needs to be balanced, within or outside of reach, initial limb configuration randomized, and unnecessary tool pick up (picking up tool when target is already within reach) needs to penalized in the performance computation.

D.2 Constructing Composite Tools for Reaching

Composite tools consist of two or more parts. Would it be possible to use NEAT to evolve controllers that can construct composite tools? Price et al. (2009) showed that Chimpanzees can be trained through observational learning to construct a simple composite tool for reaching. Fig. 8 shows Chimpanzee tool construction behavior and the experimental set up from Price et al.'s work. For our second task, we will adopt this extendable rod domain, since it is a straight-forward extension of the reaching task in Sec. D.1.

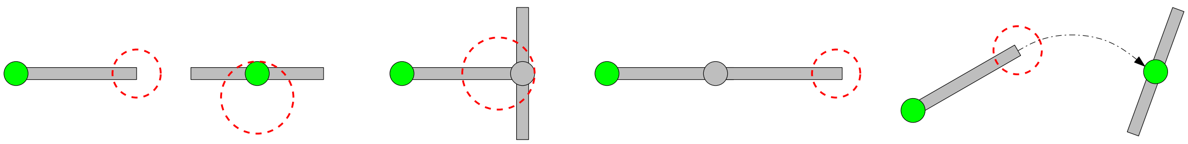
The task environment can be set up as in Sec. D.1, simply by adding an additional tool in the environment. Two instances of Tool A in Fig. 9a will be placed in the environment, so that when combined, they can become Tool A+A as shown in Fig. 9d for a longer reach. The NEAT controller network can also be extended to include a new pair of object angle and distance sensors for the second tool. The target object location will be extended so that there are now three regions: (1) region reachable without a tool, (2) region reachable with one tool (one of the two Tool As), (3) region reachable with the composite tool (Tool A+A). Based on our preliminary results (Fig. 7; Li et al. 2015), we expect this task to be solvable using an unmodified NEAT.

(a) Tool construction behavior



(b) Composite tool

Figure 8: **Tool Construction in Chimpanzees.** Chimpanzees have been shown to construct tools to extend the reach of a stick-like tool. (a) A chimpanzee putting together two rods to increase the reach. (b) Rods that can be combined to extend reach, before and after assembly, and the distance categories (C: close, MC: mid-close, MF: mid-far, F: far). The solid black rod can be sqeezed into the end of the other rod. Adapted from Price et al. (2009).



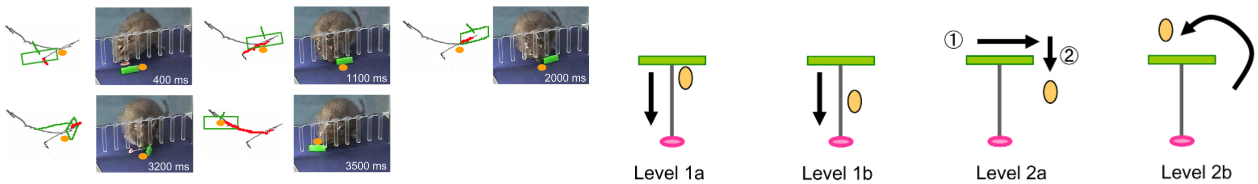
(a) Tool A (b) Tool B (c) Tool A+B (d) Tool A+A (e) Tool Construction: Tool A + Tool B

Figure 9: **Simple and Composite Tools to be Used/Constructed.** (a-b) Simple and (c-d) composite tools are shown. The green disc indicates the tool handle, the red dashed circle represents the end effector (tool tip, or the functionally effective region of the tool), and the gray disc is where the first tool's end effector joined with the second tool's tool handle. (a) Tool A can be used for reaching short-range out-of-reach objects. (b) Tool B can be used to drag within-reach objects. (c) Tool A+B is a T-shaped tool that can be used to drag out-of-reach objects. (d) Tool A+A is an extended stick-like tool for reaching long-range out-of-reach objects. (e) Tool construction process for Tool A+B. First pick up tool A, and then move its end effector toward Tool B's tool handle. The process for Tool A+A is similar. Note that functionally ineffective or ambiguous combinations are also possible, such as Tool B+A or Tool B+B which makes the tool construction task challenging.

Research Issues: The research issues in this task largely overlap with those already discussed in Sec. D.1. However, there are challenges unique to this task. One interesting challenge is to have the agents leverage on what the agents learned in Sec. D.1. That is, how can we do incremental learning? We will try two approaches. (1) Create one additional set of tool angle/distance sensors and hook them up to the output neurons directly. (2) Copy the connectivity and weights of the existing tool angle/distance sensors. Another challenge is the combinatorial nature of the tool and target placement. In Sec. D.1, if the target is out of reach, the agent just needs to pick up the tool first. However, in the composite-tool reaching task, both of the tools may be within range, or only one may be within range (the other one reachable only after picking up the closer tool). Due to this, sometimes tool 1 needs to be picked up first, while some other times tool 2 needs to be picked up first. Evolving neural networks to handle these conflicting context-dependent task requirements can be tricky, thus there is a chance that a vanilla NEAT will not work. If an unmodified NEAT does not work, we will use more sophisticated methods to be developed below (Sec. D.3-D.4).

D.3 Constructing Composite Tools for Dragging

Dragging target objects with a T-shaped tool (Fig. 9c) is a significantly more difficult task than simply reaching to touch the target (Fig. 10b), as shown in animal studies and in robotics experiments. Okanoya et al. (2008) showed that Degu (a South American rodent) can be trained to use a T-shaped rake to retrieve food located beyond a vertical barrier wall (Fig. 10). Similar experiments



(a) Rodent (Degu) Behavior (b) Dragging Tasks, Easy to Hard

Figure 10: **Dragging with a Tool.** Dragging an object is more difficult than reaching. Okanoya et al. (2008) showed that rodents (Degu) can be trained to use a rake to retrieve food located across a barrier. (a) Degu's tool use behavior and tool/target trajectory (green = rake, orange=food, red=tool trajectory). (b) Increasing levels of difficulty in the dragging task. Pink: tool handle. Green: end effector. Orange: food. Adapted from Okanoya et al. (2008).

using T-shaped and L-shaped tools were conducted with robots for dragging objects on the table (Nishide et al. 2012; Takahashi et al. 2014a; Takahshi et al. 2014b). For the robotics experiment, an elaborate dynamic sensorimotor model was needed to tackle the task.

We expect the unmodified NEAT to be able to solve the tool construction problem for this task (Fig. 9c), since the task is no different from constructing the extended tool in Sec. D.2 (Fig. 9d). (As we will discuss below in more detail in "Research Issues", if multiple tool parts are present in the environment, putting together the right tool for the task can become a challenge.)

However, for the use of the resulting T-shaped tool, NEAT may have to be modified or supplemented to deal with the more difficult task requirements. The main issues with the T-shaped tool (or "L"-shaped tools) are (1) the more complex affordance of the tool (Stoytchev 2005) and (2) multiple behavioral sequences required to achieve the goal (position tool around the target object and then drag; Fig. 10b, task Level 2a and 2b).

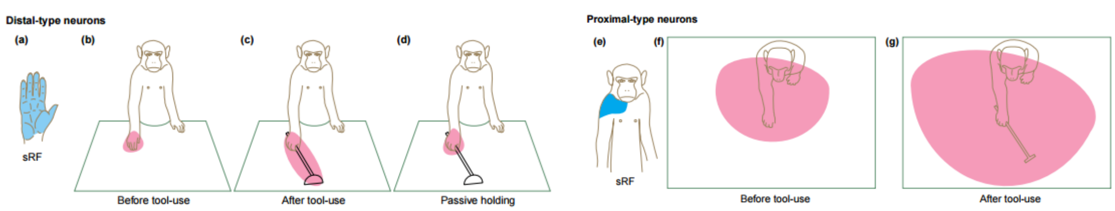
A key requirement for us in introducing additional facilities to handle these issues is *not to build in a direct solution by hand*. Thus, planning algorithms that require explicit reasoning over subtasks such as Nissim et al. (2010) or model-based probabilistic approaches such as Jain and Inamura (2014) will not be used. We will instead use (1) modular learning extension for NEAT that is able to handle interleaved and blended tasks (Schrum and Miikkulainen 2015; also see Valsalam et al. 2012), (2) a context-sensitive sensorimotor state prediction approach using recurrent neural networks (Nishide et al. 2012; Takahashi et al. 2014a). We will also try (3) a new approach we will develop for an auxiliary fitness function for indirect sub-task sequencing (details below).

A novel approach (approach 3 right above) we will develop to enable complex (indirect) planning is to introduce an auxiliary fitness function that combines fine-grained sub-fitness functions. This approach is inspired by the use of *options* (an option is a well-defined Input state set I, policy π, and terminating condition β) in hierarchical reinforcement learning (Barto and Mahadevan 2003) and *reward shaping* which allows additional rewards to be used to guide learning (Ng et al. 1999). The sub-fitness functions could be tool pickup (T), tool construction (C), tool positioning (P), dragging (D), and combining these, the auxiliary fitness function can co-evolve with the controller networks. An important feature of the auxiliary fitness function will be an evolvable ordering of the sub-task fitness functions, so that the final auxiliary fitness function computation is not commutative. For example, TCPD will not be the same as DTCP, thus multiplying the sub-fitness values will not work. Various auxiliary fitness functions will be allowed to evolve in the standard manner, and the sub-population of evolved controllers that are evaluated with the right auxiliary fitness functions will lead to higher overall fitness (the primary fitness function). This way, planning can be implicitly enforced, without explicit programming or hand-coding of the solution.

Research Issues: The task environment will now contain three kinds of objects: (1) the target object, (2) Tool A (Fig. 9a), a stick, and (3) Tool B (Fig. 9b), a rake end. Tool construction is expected to be more difficult than Sec. D.2 since the tool parts have to be picked up in the correct order (Tool A then Tool B). In principle, Tool B+A (an upside-down T-shape) should be a possible outcome, and we need to consider how to model and handle the case. What would be the end effector of the resulting tool? Also, do we need more (or different types of) sensors to detect the distance to different tool parts? One approach is to have just one set of sensors (angle and distance) to detect the closest tool part, but this will not work if Tool B is the first tool part visible. An additional sensor may be needed to discriminate the tool type. The tool construction task can be made more difficult by adding more than two tool parts in the environment. Extending the sensors in the controller network in a way not to hand-code the solution will be a challenge. Finally, we are assuming that the auxiliary fitness calculations will be based on sub-fitness values of T, C, P, and D (see above paragraph for definitions). However, these are at least indirectly forcing preconceived task knowledge. Would it be possible to discover these sub-fitness values, through affordance analysis or behavior segmentation (through clustering)? We will explore these issues as part of task D.3, and in relation to Sec. D.4.

D.4 Tool Construction and Use in Real-World Physics Simulation

So far, we focused on more abstract aspects of tool construction and use, i.e. picking up tool or tool parts in the correct order, and then using the tool to reach or drag the target object. This was made possible by setting up the inputs of the controller network at the right level of abstraction: distance and angle to the target relative to the end effector, and tool tip automatically becoming the end effector upon tool pick up, etc. However, in animals and humans, determining these seemingly simple things is a challenge in itself. What we have glossed over so far includes (1) target and tool (part) recognition and localization, (2) affordance of the tool (where's the tool handle, what function does the tool afford?, etc.; Stoytchev 2005; Thill et al. 2013) and (3) body schema and tool-body assimilation (how does one's body image change or extend when the tool is used?; Maravita and Iriki 2004; Nishide et al. 2012; Fig.11). Figs. 12a-b show a survey of existing models on body schema and affordance. In addition to the above, we need to consider (4) estimation of the spatial relationship between the end-effector and the target (Bullock et al. 1993). Finally, we have to address (5) real-world physics including friction and collision. In this task (D.4), we will address these five remaining issues.

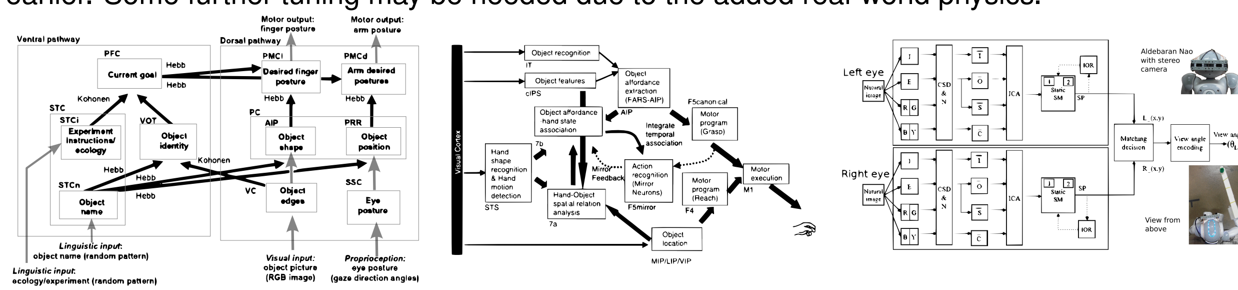


Distal Receptive Field (hand) Proximal Receptive Field (shoulder)

Figure 11: **Receptive Field Changes Due to Tool Use.** The receptive fields for distal (end effector: hand) and proximal (first joint: shoulder) parts of the arm of bimodal (visual [pink] and somatosensory [blue]) neurons are shown. The visual RFs of these neurons are known to extend due to tool use: (c) and (g). Adapted from Maravita and Iriki (2004).

To tackle these issues, we will take a step-by-step approach. (1) We will train NEAT with 2D images of the environment as input and the object type and location as target output. Instead of processing the whole image at a time, we will construct an active vision task so that a small region roughly the size of the tool (or target object) can be viewed and processed (cf. James and Tucker

2005). Together with the proprioceptive information (location currently being attended), the object category and the position can be determined and passed on to the next stage. Steps (2) and (3) will be handled together. We will combine motor gesture maps formed by GCAL (Fig. 5) and NEAT. For this, the input will again be the sequence of 2D images of the environment including the limbs, and the target output will be the limb configuration and the end-effector position. We will use image change detection to keep track of self-generated motion, including the limb and the tool (once picked up). To detect end effector location, we will use touch feedback. When the end of the limb or the tool tip touches another object, a tactile feedback will be generated. These sensory data will be used as self-supervising target values to train the affordance/body-assimilation module. (4) Once the target object and the end-effector positions are determined, we will use these as input to train a separate NEAT network to compute the angle and distance as the target output (we will also leverage our work on binocular distance estimation: Fig. 12c). (5) For the simulation of real-world physics, we will use Box2D (Catto 2013). Box2D is an open source 2D physics simulator for rigid body motion, and supports jointed objects, gravity, friction, and collision. Box2D supports both vertical 2D layouts and horizontal 2D layouts. For our task, we will use the horizontal layout (also called “top down”). Box2D simulation will be used for all steps above. Once all of the above steps are all completed, we can plug in these to the existing system developed earlier. Some further tuning may be needed due to the added real-world physics.



(a) Caligiore et al. (2010) (b) Oztop and Arbib (2002) (c) Mann et al. (2013) (PI’s work)

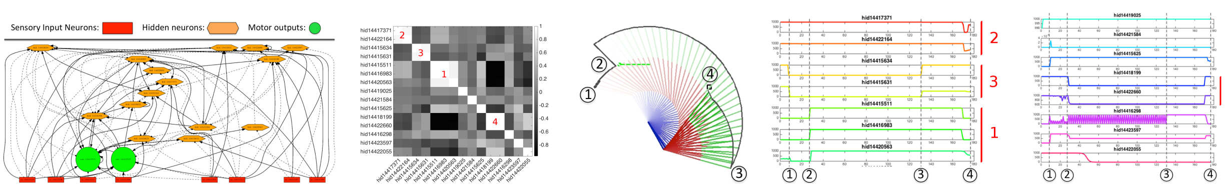
Figure 12: **Models of Schema, Affordances, and Distance Estimation.** (a)-(b) Theories on body schema and affordances. The models incorporate common themes such as visual object recognition, object localization, object affordances, motor program, action goals, etc. See Thill et al. (2013) for a comprehensive review. Note that the models shown above are descriptive, while our proposed work is constructive. (c) Distance estimation in a robot (PI’s work: Mann et al. 2013). I: intensity; E: edge; R/G & B/Y: red-green & blue-yellow opponency; CSD&N: center-surround difference and normalization; $\bar{I}, \bar{O}, \bar{S}, \bar{C}$: Intensity, orientation, symmetry, color feature map; ICA: independent components analysis; SM: saliency map; IOR: inhibition of return.

Research Issues: Object recognition and localization is expected to be relatively easy compared to other tasks, so we do not expect much issues. The remaining steps will be challenging, with multiple research issues to tackle. As we have seen in Fig. 11, the reorganization of body schema (visual RF of limb-tool) is dynamic. However, GCAL and its gesture map (Fig. 5) requires extended training. A key research issue will be to adapt GCAL so that it can dynamically adjust its gesture map based on the tool that has been picked. One possible approach is to stack two GCAL maps, one with slow dynamics (original) and the other with fast dynamics (see Nishide et al. 2012 for the combined use of fast and slow dynamics). How to incorporate change detection information for self/non-self discrimination is another issue to be resolved. Training NEAT to perform angle and distance calculation is expected to be relatively easy. Friction and collision in real-world physics simulation can pose some challenges. For example, in addition to the angle of the joints, we will also need to adjust the force. If the force is too weak compared to the mass of the limb, tool, or the target object, nothing will move. On the other hand, if the force is too strong, objects will bounce away when objects and the limb/tool collide, making the task unpredictable. So far we have not considered the interface between the hand and the tool. This is an important aspect of tool construction and use, however, given the limited budget of this proposed project (one PI and one graduate student), this will be a bit beyond our scope. However, if time permits, we wish to address the hand-tool interface. First, we will use a single arm with a gripper hand for the simple-tool reaching task. We may have to introduce the wrist and the wrist angle to make this work. Next, for tool construction, we will need two arms. From this point on, the problem becomes much more difficult to handle with the approach we took so far, as it will require a skillful coordination of two arms and hands to put together a composite tool. Finally, in nature, tool construction and use is in many cases a learned art (Price et al. 2009). Can our system be modified to facilitate observational learning through imitation? This will require learning on top of genetic search, and we will explore this possibility if time permits.

D.5 Analysis of Sensorimotor Controllers Optimized for Each Task

To gain insights into the emergence of tool use in simple controller networks, we will analyze the circuits, behavior, and internal dynamics of the evolved neural networks. The internal dynamics is initiated by input stimulus and partly modulated by the agents' own actions. Thus, analysis of internal dynamics must be investigated in the behavioral context. We will collect neuronal activation time series while the agent is behaving in the environment, so that behavior and internal activity can be co-registered.

We will take two main approaches to carry out behavior categorization (Fig. 13c). (1) Clustering: We will collect behavioral data into a multivariate vector and run cluster analysis based on multi-scale sliding windows. The resulting clusters can be used as initial behavioral categories. (2) Microstimulation: We will also conduct systematic microstimulation of the hidden neurons to elicit short yet coherent behavioral patterns. All hidden neurons will be stimulated in isolation for a varying length of time. This is inspired by the work of Graziano (2009), where he showed that short simulation of motor neurons in the macaque monkey leads to muscle twitches but prolonged duration of stimulation gives rise to complex behavioral patterns such as reaching or jumping.



(a) Evolved network (b) Activity clustering (c) Behavior (d) Neural Activity (e) Activity (Cont'd)

Figure 13: **Analysis.** A preliminary analysis of neural activity in evolved controller for the reaching task is shown. (a) Evolved network with 15 hidden neurons (orange hexagons), 8 input neurons (red rectangles at the bottom) and 2 output neurons (green circles). (b) Clustering of hidden neurons based on their activity. (c) Reaching behavior shown as a time-lapsed image (①: reverse direction; ②: pick up tool; ③: reverse direction; ④: target reached). (d)-(e): Activity in the hidden neurons during behavior shown in (c). At the bottom, circled numbers indicate the behavioral events in (c). Numbered rows along the vertical axes (in red) show the clusters in (b). From PI's recent work (Li et al. 2015).

Analysis of the internal dynamics (Fig. 13d-e) will be in many ways similar to that of the motor behavior: (1) clustering (Fig. 13b), (2) segmentation, (3) phase-plane analysis (including detection of phase transitions), and (4) stimulus-induced activity measurement to characterize receptive fields. Furthermore, (5) we will extract time series data that correspond to component actions identified in the behavioral analysis and align their properties (Fig. 13d-e). (6) We will also check the correlation between the number of cycles in the neural circuit (the connectivity graph), as these recurrent loops contribute to memory and give rise to periodic dynamics (Fig. 14). Finally, (7) we will conduct selected microstimulations and observe their effects. For the analysis, we will use mutual information (Sporns and Tononi 2002), predictability (Choe et al. 2012; Kwon and Choe 2008), synchronization (Choe and Miikkulainen 2004), and other quantitative measures that are known to be important indicators of functional principles.

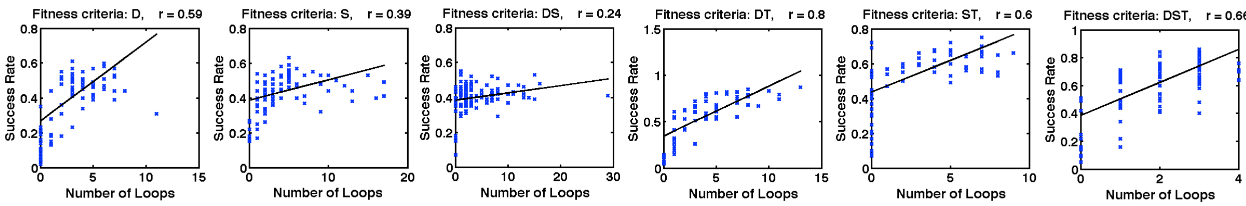


Figure 14: **Number of Cycles in the Connectivity Graph vs. Performance.** The correlation between number of cycles (loops) in the neural circuit graph vs. fitness is shown for different fitness criteria used (D: distance to target; S: # steps taken; T: # times tool picked up; DS, ST, DST: combined fitness [by multiplication]). The task was the reaching task in Figure 6. More cycles generally led to higher fitness. Adapted from preliminary results reported in Li et al. (2015).

Research issues: Neural network controllers with 3 or fewer hidden neurons are easy to visualize. However, in a general case, the internal state has a much higher dimension. Our planned clustering analysis can help, but we will also use dimensionality reduction techniques, both linear (PCA) and nonlinear (ISOMAP, Locally Linear Embedding, etc.) to tackle this issue. In terms of counting the loops, we may have to normalize the count by the overall size of the graph (number of nodes), since more advanced circuits appear later in the generation and they tend to be large in size and thus potentially contain more loops. Finally, since connections can have either a positive sign (excitatory) or a negative sign (inhibitory), the analysis can become non-trivial. Almost all graph-based approaches assume that connections are excitatory. With inhibitory connections, disinhibition and other unintuitive effects can emerge (see e.g., Choe 2004). We expect this difficulty can be overcome to some degree by considering groups of neurons as a unit and minimizing inhibitory interaction across groups. Finally, to gain insights on tool contruction and tool use, we will conduct a comparative analysis of the circuits and functional modules with existing models of body schema and affordances (Fig. 12a-b).

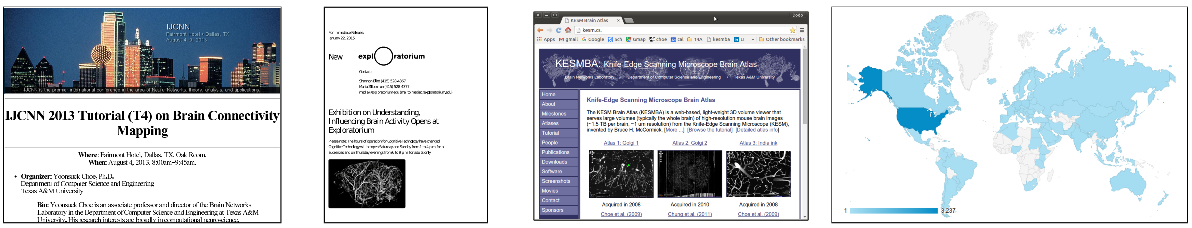
E Broader Impacts of the Proposed Work

Interactive web-portal for the exploration of evolved agent and its behavior: This research project will generate a variety of simple to complex neural controllers and simulations in sensorimotor control and tool construction/use tasks. These neural circuits and behavioral simulations will be curated and integrated into an online portal where the general public can interact, experiment, and analyze those circuits and simulations. The research team will publicize the resource through YouTube videos and social media.

Tool construction and use challenge: A "tool construction and use challenge" will be organized at major conferences such as International Joint Conference on Artificial Intelligence and World Congress on Computational Intelligence. We will develop a simulation engine and a client-server protocol for use in the competition.

Education: Graduate students will be trained as part of this project (direct funding via this grant). Undergraduate students will be trained using the REU supplement mechanism (support for year 1 [summer 2017] requested in the budget). Research results will be incorporated into graduate and undergraduate level courses: Introduction to Artificial Intelligence (undergrad and graduate), Machine Learning (graduate), Neural Networks (graduate), and Cortical Networks (graduate).

Dissemination: Once published in peer-review venues, all source code and data will be released to the public using GNU Public License and Open Data Commons Attribution License. See the Data Management Plan in the supplementary documents section for details. We will also organize tutorials and exhibits at conferences (see Fig. 15 for our prior efforts).



(a) Tutorial (b) Museum Exhibit (c) KESM Brain Atlas (d) World-wide access

Figure 15: **Broader Impact (Outreach).** Web page screenshots: (a) 2013 Int'l Joint Conference on Neural Networks Tutorial on connectivity mapping. (b) 2015 San Francisco Exploratorium Exhibit on the brain (featuring KESM India ink data at the bottom; exhibit arranged by collaborting startup company 3Scan, located in San Francisco). (c) KESM Brain Atlas. Google Analytics data since February 2012: 4,727 visitors, 13,081 page views. (d) World-wide access plot (Google Analytics). Some screenshots were edited (elements rearranged) to fit the page.

Summary of Broader Impacts: The interactive portal will allow the general public to engage in the project in a playful and entertaining manner, all the while being educated. The tool construction and use challenge will start a whole new field, and the collective findings will help propel AI and robotics research. Graduate and undergraduate students trained in this project are expected to become experts in modeling sensorimotor learning. Code and data dissemination will enable other research groups to easily extend upon this research.

F Management Plan

The time line for each task is shown below. The PI and the graduate student will be the core personnel for this project. Each semester, roughly 2 to 3 tasks are scheduled. The team will hold a weekly meeting to make continual progress. Broader impact activities (REU training and dissemination) are included in all 3 years.

Table 1: Tasks and Timeline

Task	Y1			Y2			Y3		
D.1 Using Simple Tools for Reaching	—	—							
D.2 Constructing Composite Tools for Reaching		—	—	—					
D.3 Constructing Composite Tools for Dragging			—	—	—	—			
D.4 Tool Construction & Usage in Real-World Physics Simulation					—	—	—	—	—
D.5 Analysis of Sensorimotor Controllers Optimized for Each Task			—			—		—	—
E. Broader Impact			—			—			—

REFERENCES CITED

Agogino, A., Tumer, K., and Miikkulainen, R. (2005). Efficient credit assignment through evaluation function decomposition. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, 1309–1316. ACM.

Allen, C. (2006). Transitive inference in animals: reasoning or conditioned associations. *Rational animals*, 175–185.

Ambrose, S. H. (2001). Paleolithic technology and human evolution. *Science*, 291(5509):1748–1753.

Arsenio, A. M. (2004). Learning task sequences from scratch: applications to the control of tools and toys by a humanoid robot. In *Control Applications, 2004. Proceedings of the 2004 IEEE International Conference on*, vol. 1, 400–405. IEEE.

Auersperg, A., von Bayern, A., Weber, S., Szabadvari, A., Bugnyar, T., and Kacelnik, A. (2014). Social transmission of tool use and tool manufacture in goffin cockatoos (cacatua goffini). *Proceedings of the Royal Society B: Biological Sciences*, 281(1793):20140972.

Barto, A. G., and Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 13(1-2):41–77.

Bednar, J. A. (2012). Building a mechanistic model of the development and function of the primary visual cortex. *Journal of Physiology-Paris*, 106(5-6):194–211.

Boesch, C., and Boesch, H. (1990). Tool use and tool making in wild chimpanzees. *Folia primatologica*, 54(1-2):86–99.

Bullock, D., Grossberg, S., and Guenther, F. H. (1993). A self-organizing neural model of motor equivalent reaching and tool use by a multijoint arm. *Journal of Cognitive Neuroscience*, 5(4):408–435.

Caligiore, D., Borghi, A. M., Parisi, D., and Baldassarre, G. (2010). Tropicals: a computational embodied neuroscience model of compatibility effects. *Psychological Review*, 117(4):1188.

Catto, E. (2013). Box2d v2.3.0 user manual. http://www.box2d.org/manual.pdf.

Choe, Y. (2004). The role of temporal parameters in a thalamocortical model of analogy. *IEEE Transactions on Neural Networks*, 15:1071–1082.

Choe, Y., Kwon, J., and Chung, J. R. (2012). Time, consciousness, and mind uploading. *International Journal on Machine Consciousness*, 4:257–274.

Choe, Y., Mayerich, D., Kwon, J., Miller, D. E., Chung, J. R., Sung, C., Keyser, J., and Abbott, L. C. (2011a). Knife-edge scanning microscopy for connectomics research. In *Proceedings of the International Joint Conference on Neural Networks*, 2258–2265. Piscataway, NJ: IEEE Press.

Choe, Y., Mayerich, D., Kwon, J., Miller, D. E., Sung, C., Chung, J. R., Huffman, T., Keyser, J., and Abbott, L. C. (2011b). Specimen preparation, imaging, and analysis protocols for knife-edge scanning microscopy. *Journal of Visualized Experiments*, 58:e3248. doi: 10.3791/3248.

Choe, Y., and Miikkulainen, R. (2004). Contour integration and segmentation in a self-organizing map of spiking neurons. *Biological Cybernetics*, 90:75–88.

Choe, Y., Yoo, J., and Li, Q. (2015). Tool construction and use challenge: Tooling test rebooted. In *AAAI-15 Workshop on Beyond the Turing Test*. 2 pages.

Choi, J. (2013). *Knife-Edge Scanning Microscope Mouse Brain Atlas in Vector Graphics for Enhanced Performance*. Master's thesis, Department of Computer Science and Engineering, Texas A&M University.

ChrisO (2005). Counterweight trebuchet at Château des Baux, France. `https://commons.wikimedia.org/wiki/File:Trebuchet.jpg`.

Chung, J. R. (2011). *Evolution of Memory in Reactive Artificial Neural Networks*. PhD thesis, Department of Computer Science, Texas A&M University.

Chung, J. R., and Choe, Y. (2009). Emergence of memory-like behavior in reactive agents using external markers. In *Proceedings of the 21st International Conference on Tools with Artificial Intelligence, 2009. ICTAI '09*, 404–408.

Chung, J. R., and Choe, Y. (2011a). Emergence of memory in reactive agents equipped with environmental markers. *Autonomous Mental Development, IEEE Transactions on*, 3(3):257–271.

Chung, J. R., and Choe, Y. (2011b). Emergence of memory in reactive agents equipped with environmental markers. *IEEE Transactions on Autonomous Mental Development*, 3:257–271.

Chung, J. R., Kwon, J., and Choe, Y. (2009). Evolution of recollection and prediction in neural networks. In *Proceedings of the International Joint Conference on Neural Networks*, 571–577. Piscataway, NJ: IEEE Press.

Chung, J. R., Sung, C., Mayerich, D., Kwon, J., Miller, D. E., Huffman, T., Abbott, L. C., Keyser, J., and Choe, Y. (2011). Multiscale exploration of mouse brain microstructures using the knife-edge scanning microscope brain atlas. *Frontiers in Neuroinformatics*, 5:29.

Chung, M. J.-Y., Friesen, A. L., Fox, D., Meltzoff, A. N., and Rao, R. P. (2015). A bayesian developmental approach to robotic goal-based imitation learning. *PloS one*, 10(11):e0141965.

Cowley, M. (1996). *Dar and the Spear Thrower*. Houghton Mifflin Harcourt.

Dileepkumar, A. (2014). *Semi-Automated Reconstruction of Vascular Networks in Knife-Edge Scanning Microscope Mosue Brain Data*. Master's thesis, Department of Computer Science and Engineering, Texas A&M University.

Foerder, P., Galloway, M., Barthel, T., Moore III, D. E., and Reiss, D. (2011). Insightful problem solving in an asian elephant. *PloS one*, 6(8):e23251.

Fullmer, B., and Miikkulainen, R. (1992). Using marker-based genetic encoding of neural networks to evolve finite-state behaviour. In Varela, F. J., and Bourgine, P., editors, *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, 255–262. Cambridge, MA: MIT Press.

Gomez, F., and Miikkulainen, R. (1997). Incremental evolution of complex general behavior. *Adaptive Behavior*, 5(3-4):317–342.

Graziano, M. (2009). *The Intelligent Movement Machine: An Ethological Perspective on the Primate Motor System*. Oxford, UK: Oxford University Press.

Graziano, M. S., Taylor, C. S., and Moore, T. (2002). Complex movements evoked by microstimulation of precentral cortex. *Neuron*, 34:841–851.

Gumert, M. D., Kluck, M., and Malaivijitnond, S. (2009). The physical characteristics and usage patterns of stone axe and pounding hammers used by long-tailed macaques in the andaman sea region of thailand. *American Journal of Primatology*, 71(7):594–608.

Hall, K. R. L. (1963). Tool-using performances as indicators of behavioral adaptability. *Current Anthropology*, 479–494.

Jain, R., and Inamura, T. (2014). Learning of usage of tools based on interaction between humans and robots. In *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2014 IEEE 4th Annual International Conference on*, 597–602. IEEE.

James, D., and Tucker, P. (2005). Evolving a neural network active vision system for shape discrimination. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2005) Late Breaking Papers*.

Katz, D., and Brock, O. (2008). Manipulating articulated objects with interactive perception. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 272–277. IEEE.

Kenward, B., Weir, A. A., Rutz, C., and Kacelnik, A. (2005). Behavioural ecology: Tool manufacture by naive juvenile crows. *Nature*, 433(7022):121–121.

Kim, D. (2011). *Automatic Seedpoint Selection and Tracing of Microstructures in the Knife-Edge Scanning Microscope Mouse Brain Data Set*. Master's thesis, Department of Computer Science, Texas A&M University, College Station, Texas.

Kohler, W. (1924). *The mentality of apes*. Harcourt, Brace.

Kwon, J., and Choe, Y. (2008). Internal state predictability as an evolutionary precursor of self-awareness and agency. In *Proceedings of the Seventh International Conference on Development and Learning*, 109–114. IEEE.

Kwon, J., Mayerich, D., and Choe, Y. (2011). Automated cropping and artifact removal for knife-edge scanning microscopy. In *Proceedings of the IEEE International Symposium on Biomedical Imaging*, 1366–1369.

Lal Das, S. (2014). *Cell Detection in Knife-Edge Scanning Microscopy Images of Nissl-stained Mouse and Rat Brain Samples using Random Forests*. Master's thesis, Department of Computer Science and Engineering, Texas A&M University.

Lal Das, S., Keyser, J., and Choe, Y. (2015). Random-forest-based automated cell detection in knife-edge scanning microscope rat nissl data. In *Proceedings of the International Joint Conference on Neural Networks*. DOI: 10.1109/IJCNN.2015.7280852.

Law, J. S., Antolik, J., and Bednar, J. A. (2011). Mechanisms for stable and robust development of orientation maps and receptive elds. Technical report, School of Informatics, The University of Edinburgh, EDI-INF-RR-1404.

Lee, D., Kunori, H., and Nakamura, Y. (2008). Association of whole body motion from tool knowledge for humanoid robots. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 2867–2874. IEEE.

Li, Q., Yoo, J., and Choe, Y. (2015). Emergence of tool use in an articulated limb controlled by evolved neural circuits. In *Proceedings of the International Joint Conference on Neural Networks*. DOI: 10.1109/IJCNN.2015.7280564.

Mann, T. A., and Choe, Y. (2010). Prenatal to postnatal transfer of motor skills through motor-compatible sensory representations. In *Development and Learning (ICDL), 2010 IEEE 9th International Conference on*, 185–190. IEEE.

Mann, T. A., Park, Y., Jeong, S., Lee, M., and Choe, Y. (2013). Autonomous and interactive improvement of binocular visual depth estimation through sensorimotor interaction. *IEEE Transactions on Autonomous Mental Development*, 5:74–84.

Maravita, A., and Iriki, A. (2004). Tools for the body (schema). *Trends in cognitive sciences*, 8(2):79–86.

Mayerich, D., Kwon, J., Panchal, A., Keyser, J., and Choe, Y. (2011a). Fast cell detection in high-throughput imagery using gpu-accelerated machine learning. In *Proceedings of the IEEE International Symposium on Biomedical Imaging*, 719–723.

Mayerich, D., Kwon, J., Sung, C., Abbott, L. C., Keyser, J., and Choe, Y. (2011b). Fast macro-scale transmission imaging of microvascular networks using KESM. *Biomedical Optics Express*, 2:2888–2896.

Miikkulainen, R., Bednar, J. A., Choe, Y., and Sirosh, J. (2005). *Computational Maps in the Visual Cortex*. Springer.

Miller, D. E. (2014). *A Combined Skeleton Model*. Master's thesis, Department of Computer Science and Engineering, Texas A&M University.

Montana, D. J., and Davis, L. (1989). Training feedforward neural networks using genetic algorithms. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, 762–767. San Francisco, CA: Morgan Kaufmann.

Moriarty, D. E., and Miikkulainen, R. (1997). Forming neural networks through efficient and adaptive co-evolution. *Evolutionary Computation*, 5:373–399.

Murphy, R. (2000). *Introduction to AI robotics*. MIT press.

Ng, A. Y., Harada, D., and Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, vol. 99, 278–287.

Nishide, S., Tani, J., Takahashi, T., Okuno, H. G., and Ogata, T. (2012). Tool–body assimilation of humanoid robot using a neurodynamical system. *Autonomous Mental Development, IEEE Transactions on*, 4(2):139–149.

Nissim, R., Brafman, R. I., and Domshlak, C. (2010). A general, fully distributed multi-agent planning algorithm. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, 1323–1330. International Foundation for Autonomous Agents and Multiagent Systems.

Okanoya, K., Tokimoto, N., Kumazawa, N., Hihara, S., and Iriki, A. (2008). Tool-use training in a species of rodent: the emergence of an optimal motor strategy and functional understanding. *PLoS One*, 3(3):e1860.

Onno (2008). Operation scheme of a trebuchet. `https://commons.wikimedia.org/wiki/File:Trebuchet_Scheme.png`.

Oztop, E., and Arbib, M. A. (2002). Schema design and implementation of the grasp-related mirror neuron system. *Biological cybernetics*, 87(2):116–140.

Pastor, P., Hoffmann, H., Asfour, T., and Schaal, S. (2009). Learning and generalization of motor skills by learning from demonstration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, 763–768. IEEE.

Potter, M. A., and De Jong, K. A. (2000). Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8:1–29.

Price, E. E., Lambeth, S. P., Schapiro, S. J., and Whiten, A. (2009). A potent effect of observational learning on chimpanzee tool construction. *Proceedings of the Royal Society of London B: Biological Sciences*, rspb20090640.

Ruiz, C. (2014). Primitive fire making 6 ways to make fire without matches. `http://www.thebugoutbagguide.com/2014/09/24/primitive-fire-making/`.

Saegusa, R., Metta, G., Sandini, G., and Natale, L. (2014). Developmental perception of the self and action. *Neural Networks and Learning Systems, IEEE Transactions on*, 25(1):183–202.

Schäfer, B., Bergfeldt, N., Riveiro Carballa, M. J., and Ziemke, T. (2007). Evolution of tool use behavior. In *Proceedings of the First IEEE Symposium on Artificial Life*, 31–38. Citeseer.

Schrum, J., and Miikkulainen, R. (2015). Solving interleaved and blended sequential decision-making problems through modular neuroevolution. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2015)*, 345–352. Madrid, Spain. Best Paper: Digital Entertainment and Arts.

Shah, R. S. (2014). *Reducing Chatter in Knife-Edge Scanning Microscopy*. Master's thesis, Department of Computer Science and Engineering, Texas A&M University.

Shumaker, R. W., Walkup, K. R., and Beck, B. B. (2011). *Animal tool behavior: the use and manufacture of tools by animals*. JHU Press.

Sims, K. (1994). Evolving 3d morphology and behavior by competition. *Artificial life*, 1(4):353–372.

Singhal, A. (2015). *Skeletonization-Based Automated Tracing and Reconstruction of Neurovascular Networks in Knife-Edge Scanning Microscope Mouse Brain India Ink Data*. Master's thesis, Department of Computer Science and Engineering, Texas A&M University.

Smolker, R., Richards, A., Connor, R., Mann, J., and Berggren, P. (1997). Sponge carrying by dolphins (delphinidae, tursiops sp.): a foraging specialization involving tool use? *Ethology*, 103(6):454–465.

Snowdon, C. T. (1990). Language capacities of nonhuman animals. *American Journal of Physical Anthropology*, 33(S11):215–243.

Sporns, O., and Tononi, G. (2002). Classes of network connectivity and dynamics. *Complexity*, 7:28–38.

Srivastava, M. (2015). *Knife-Edge Scanning Microscope Brain Atlas Interface for Tracing and Analysis of Vasculature Data*. Master's thesis, Department of Computer Science and Engineering, Texas A&M University.

St. Amant, R., and Wood, A. B. (2005). Tool use for autonomous agents. In *Twenteth AAAI Conference on Artificial Intelligence*, 184–189.

Stanley, K. O., and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:99–127.

Stoytchev, A. (2005). Behavior-grounded representation of tool affordances. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 3060–3065. IEEE.

Sung, C. (2013). *Exploration, Registration, and Analysis of High-Throughput 3D Microscopy Data from the Knife-Edge Scanning Microscope*. PhD thesis, Department of Computer Science and Engineering, Texas A&M University.

Sung, C., Woo, J., Goodman, M., Huffman, T., and Choe, Y. (2013). Scalable, incremental learning with MapReduce parallelization for cell detection in high-resolution 3D microscopy data. In *Proceedings of the International Joint Conference on Neural Networks*, 434–440.

Takahashi, K., Ogata, T., Tjandra, H., Murata, S., Arie, H., and Sugano, S. (2014a). Tool-body assimilation model based on body babbling and a neuro-dynamical system for motion generation. In *Artificial Neural Networks and Machine Learning–ICANN 2014*, 363–370. Springer.

Takahshi, K., Ogata, T., Tjandra, H., Yamaguchi, Y., Suga, Y., and Sugano, S. (2014). Tool-body assimilation model using a neuro-dynamical system for acquiring representation of tool function and motion. In *Advanced Intelligent Mechatronics (AIM), 2014 IEEE/ASME International Conference on*, 1255–1260. IEEE.

Takahshi, K., Ogata, T., Tjandra, H., Yamaguchi, Y., Suga, Y., and Sugano, S. (2014b). Tool-body assimilation model using a neuro-dynamical system for acquiring representation of tool function and motion. In *Advanced Intelligent Mechatronics (AIM), 2014 IEEE/ASME International Conference on*, 1255–1260. IEEE.

Thill, S., Caligiore, D., Borghi, A. M., Ziemke, T., and Baldassarre, G. (2013). Theories and computational models of affordance and mirror systems: an integrative review. *Neuroscience & BioBehavioral Reviews*, 37(3):491–521.

Valsalam, V. K., Hiller, J., MacCurdy, R., Lipson, H., and Miikkulainen, R. (2012). Constructing controllers for physical multilegged robots using the enso neuroevolution approach. *Evolutionary Intelligence*, 5(1):45–56.

Wang, L., Brodbeck, L., and Iida, F. (2014). Mechanics and energetics in tool manufacture and use: a synthetic approach. *Journal of The Royal Society Interface*, 11(100):20140827.

Whiten, A., Goodall, J., McGrew, W. C., Nishida, T., Reynolds, V., Sugiyama, Y., Tutin, C. E., Wrangham, R. W., and Boesch, C. (1999). Cultures in chimpanzees. *Nature*, 399(6737):682–685.

Whitley, D., Dominic, S., Das, R., and Anderson, C. W. (1993). Genetic reinforcement learning for neurocontrol problems. *Machine Learning*, 13:259–284.

Wieland, A. P. (1990). Evolving controls for unstable systems. In Touretzky, D. S., Elman, J. L., Sejnowski, T. J., and Hinton, G. E., editors, *Connectionist Models: Proceedings of the 1990 Summer School*, 91–102. San Francisco, CA: Morgan Kaufmann.

Wu, Y., and Demiris, Y. (2011). Learning dynamical representations of tools for tool-use recognition. In *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, 2664–2669. IEEE.

Yang, H.-F. (2011). *Reconstruction of 3D Neuronal Structures from Densely Packed Electron Microscopy Data Stacks*. PhD thesis, Department of Computer Science and Engineering, Texas A&M University.

Yang, H.-F., and Choe, Y. (2010). Electron microscopy image segmentation with estimated symmetric three-dimensional shape prior. In *Proceedings of the 6th International Symposium on Visual Computing*.

Yang, H.-F., and Choe, Y. (2011a). Ground truth estimation by maximizing topological agreements in electron microscopy data. In *Proceedings of the 7th International Symposium on Visual Computing (LNCS 6938)*, 371–380.

Yang, H.-F., and Choe, Y. (2011b). An interactive editing framework for electron microscopy image segmentation. In *Proceedings of the 7th International Symposium on Visual Computing (LNCS 6938)*, 400–409.

Yang, W. (2014). *Automated neurovascular tracing and analysis of the Knife-Edge Scanning Microscope India ink data set*. Master's thesis, Department of Computer Science and Engineering, Texas A&M University.

Yao, X. (1999). Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423–1447.

Yoo, J., Choi, J., and Choe, Y. (2014). Development of target reaching gesture map in the cortex and its relation to the motor map: A simulation study. In *Advances in Self-Organizing Maps and Learning Vector Quantization: Proceedings of the 10th International Workshop, WSOM 2014, Mittweida, Germany, July 2–4, 2014*, 187–197. Heidelberg: Springer.

Zhang, W. (2014). *Real-time Image Error Detection in Knife-Edge Scanning Microscope*. Master's thesis, Department of Computer Science and Engineering, Texas A&M University.

Zhang, W., Yoo, J., Keyser, J., Abbott, L. C., and Choe, Y. (2015). Real-time detection of imaging errors in the knife-edge scanning microscope through change detection. In *Proceedings of the IEEE International Symposium on Biomedical Imaging*. Accepted.