8-28-2012

A new class of neural architectures to model episodic memory : computational studies of distal reward learning

Shawn Taylor

Shawn Taylor

Candidate

ECE

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Thomas Caudell , Chairperson

Gregory Heileman

Vincent Calhoun

Derek Hamilton

A New Class of Neural Architectures to Model Episodic Memory: Computational Studies of Distal Reward Learning

by

Shawn E. Taylor

B.S.E.E., New Mexico Institute of Mining and Technology, 2001
M.S.E.E., University of New Mexico, 2006

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Engineering

The University of New Mexico
Albuquerque, New Mexico

July, 2012

Dedication

To my children: may the ever expanding work of developing academics, of which this contribution is but a tiny part, make it possible to experience your world with greater understanding.

Acknowledgments

I would like to thank my advisor, Professor Thomas Caudell, for his support and guidance. I would also like to thank the sources of financial support through my graduate eduction; the National Science Foundation, the Defense Threat Reduction Agency, and Sandia National Labs. Numerous people have been instrumental in making the details of my relationships with those entities work. A broad stroke is unfair for all the people who enabled that support, without you nothing would get done. Moving into the realm of professional and academic colleagues, I would like to thank Michael Bernard for opportunities and insights, and Howard Eichenbaum, and Neal Cohen for insights that enriched this research greatly. My dissertation committee, including Michael Healy, deserves praise for the work that has gone into enabling this to be a successful dissertation. Finally, thank you to my wife for putting up with an apparently never ending stream of work that demanded my time.

A New Class of Neural Architectures to Model Episodic Memory: Computational Studies of Distal Reward Learning

by

Shawn E. Taylor

B.S.E.E., New Mexico Institute of Mining and Technology, 2001
M.S.E.E., University of New Mexico, 2006
PhD, Engineering, University of New Mexico, 2012

Abstract

A computational cognitive neuroscience model is proposed, which models episodic memory based on the mammalian brain. A computational neural architecture instantiates the proposed model and is tested on a particular task of distal reward learning. Categorical Neural Semantic Theory informs the architecture design. To experiment upon the computational brain model, embodiment and an environment in which the embodiment exists are simulated. This simulated environment realizes the Morris Water Maze task, a well established biological experimental test of distal reward learning. The embodied neural architecture is treated as a virtual rat and the environment it acts in as a virtual water tank. Performance levels of the neural architectures are evaluated through analysis of embodied behavior in the distal reward learning task. Comparison is made to biological rat experimental data, as well as comparison to other published models. In addition, differences in performance are compared between the normal and categorically informed versions of the architecture.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In cognitive neuroscience, at the boundary between neurophysiology and behavior, there lies one of the great unknown regions in the space of modern scientific knowledge. The human brain contains roughly 10^{11} neurons and 10^{14} synapses [3]. The magnitude of human behavior is much harder to quantify than are neurons and synapses to count, but can be appreciated by considering the distance from heights of intellectual accomplishment to depths of under-achievement; from compassionate humanitarian service to patriotic nationalism to material self-centeredness; between every emotion written about in every novel and all the facets of human behavior witnessed through history. The behaviors alone would require uncountably numerous terms to describe, and the possible mappings from neuro-structure to behavior are combinatorially larger. All of that behavior and more is a product of the embodied human nervous system by way of evolution and development in its environment. A challenge still unsolved is to understand the mapping from the functioning of all those neurons and synapses to those numerous behaviors. Neurophysiology, cognitive psychology, computational neuroscience, and brain imaging will all play a role in unraveling this challenge. The contribution of this interdisciplinary dissertation is

a new neuroscience based computational model of a "cognitive" system. It is posited that a biophysically motivated neural architecture which models episodic memory enabling aspects of the cortical-hippocampal system from mammalian brains will have the capacity, when embodied in an agent, to exhibit distal reward behavior. This model links the form of the developed neural architecture to the function of distal reward behavior. Furthermore, a nascent mathematical theory of neural structure and function is employed during the design process and its effects on behavioral performance are assessed.

1.2 Overview

The following section gives a high level description of the research question and approach. First, a statement of the problem being addressed is given. Then, potential methods for addressing the problem are described. Next, an overview of the approach used in this dissertation is provided. Finally, an outline of the rest of the dissertation appears.

1.2.1 Problem Statement

At the highest level, the problem addressed by this dissertation is to study the transformation between neural form and embodied behavior. Distal reward learning is the function that is investigated to study form to function mapping. Distal reward learning is the process by which an animal associates a reward in the present with past behaviors that lead up to that reward.. The Background chapter provides an in depth look at distal reward learning but a short introduction is given here to provide context for the initial introduction of subsequent topics which address or involve distal reward learning.

In the context of biological organisms, distal reward problems require animals to learn to correct previous decisions from future rewards. That is, from the reference

frame of the rewards the decisions are previous, from the reference frame of the decisions the rewards are future. The term distal refers to temporal distance between action and reward. Pushing a button to receive a reward is not a distal reward task if reward for a correct button push occurs immediately. Navigation is in general a distal reward task, specifically if the navigational goal is not observable from locations where navigation decisions must be made. One of the challenges for a neural system to perform distal reward learning is that the neural activations which were responsible for attaining reward may be gone by the time the reward occurs.

1.2.2 Proposed Architecture

This section presents a high level view of a new computational neural architecture which will be used in the experimental section of this work. The Background chapter will present research results from the literature, and link these to the proposed computational neural architecture. To set the context for the upcoming discussion, an introduction to the architecture is given below. The Apparatus chapter will describe the architecture in full detail.

Described at a high level, the architecture operates by chaining expected reward maximizing actions through time and space. The temporal aspect of chaining, one location in time followed by another, enables distal reward learning. The spatial aspect of chaining, one location in space followed by another, enables navigation. Together, the enabled behavior is performance of a distal reward navigation task. Expected reward level is defined in the context of reaching a goal in the environment in which an architecture is embodied. The only inputs on which the architecture can base its expected reward maximizing decisions are the sensory inputs of the simulated embodiment.

Sequences of sensory input patterns are bound together as memories or experience episodes. Note that we use the term "bound" in the sense of a set of activations being bound into the neural substrate, i.e. a memory formed. Expected reward

levels are encoded with the memories of experienced sensory episodes. For instance, an episode which includes reaching the goal will have the highest expected reward level. It is expected that if the same episode is experienced again, the same reward will be experienced again. Hence, if learned memories indicate that reward has in the past immediately followed a known series of actions performed from the current location, expected reward will be at the highest level for recall of that series of actions. An alternate series of actions, that did not have reward attainment bound in an associated memory, might have zero expected reward. Chaining begins when an episode is experienced which results in observation of the beginning of an episode which leads to reward. The episode once removed from the episode which leads to reward has a proportionately lower expected reward than the episode leading directly to reward. Through this discounting strategy, given a choice among action sequences, one will be picked which leads more directly to the reward. Thus a form of optimal behavior can occur which provides the embodied agent with a competitive advantage in expending less energy reaching reward as compared to a less direct strategy. This process addresses the distal reward problem by circumventing the need to assign reward credit back through a temporal sequence after the reward has been received. Instead, reward credit propagates through the sequence as it forms.

1.2.3 Dissertation Research Elements

As part of the Overview, some notable elements of this research are introduced here. Starting with the first experiments performed for this research [4], and continuing through to the final theoretically driven architectural modification, the Categorical Neural Semantics Theory, based in category theory, was employed for the insight it provides into structure based study of neural architectures. Episodic memory is studied and computationally modeled as a method to address the distal reward problem. Neuroscience and cognitive psychology were extensively surveyed to provide a foundation for the synthesis of the new architecture. The neuroscience and cognitive

psychology literature were examined for insight into episodic memory and a brain area called hippocampus which support episodic memory function. Computational embodiment is another notable element of this research, it provides a perceptual basis for inputs to neural architectures.

Category theory, as the mathematical discipline concerned with mathematical structure, informs this research by adding rigor to the analysis and construction of neural structure. Category theory motivates a fundamental design change to reflect a certain type of categorical structure that we hypothesize will improve embodied performance.

As distal reward behavior involves temporal processing of neural information, and episodic memory involves the encoding of events through time, episodic memory based behavior is a natural avenue of research when the goal is to understand distal reward behavior. Hippocampus is a biological brain structure for which evidence is reviewed of strong involvement in episodic memory based behavior. Hippocampus is located at the edge of cortex where almost all sensory cortices converge. This convergence as well as other high degrees of connectivity position hippocampus for a central role in behavior related to episodic memory and distal reward. As such, knowledge of hippocampus form and function is pursued in order to develop the architecture.

Computational embodiment is a key characteristic of this research. Stand-alone, non-embodied experiments provide input to this dissertation research, but the final experimental test is embodied behavior. Embodiment is key because it gives the neural architecture semantic meaning through perceptual grounding . Dreyfus presents an argument for embodiment as a necessity of general artificial intelligence [2]. General artificial intelligence is also known as strong AI and is the ability to perform any intellectual task as well or better than a human. This dissertation does not claim to address general artificial intelligence, but by conducting experiments in simulated full embodiment, we operate in the same context which gave rise to human

intelligence.

1.2.4 Document Structure

This dissertation document is structured as Introduction, Background, Apparatus, Approach, Results, Discussion, Summary and Conclusion, with the last chapter covering potential future work. The Background chapter will give neuro-anatomical, physiological, and psychological motivation and knowledge which pertain to distal reward behavior. The Background will also introduce the Categorical Neural Semantic Theory and review other models from the literature which are comparable to what was develop for this dissertation research. The Apparatus chapter will describe the details of the neural architectures which will be employed for experimentation. As well, the Apparatus chapter will detail the implementation details by which a neural architecture is embodied, and the implementation details by which an environment is simulated for the embodiment. The Approach chapter then describes the experiments which will be performed and the experimental parameters which will be applied. The Results chapter provides data produced by carrying out the experiments described in Approach. The Discussion chapter analyzes the results in the context of the dissertation objectives. The Discussion also compares and contrasts the results obtained here with the results of other comparable models. The Summary and Conclusion chapter will finish this dissertation, providing an overview of the objectives, experiments carried out to accomplish those objectives, results, and conclusions following from those results. The Conclusion also lays out the scientific contribution produced through this research. Following the Conclusion, the Future Work chapter extrapolates follow on endeavors that would be valuable further extensions to this research.

This chapter has introduced the research undertaken for this dissertation. The introduction included motivation for why this research was performed, a statement of the problem being addressed, an overview of the research elements, then an outline

of the rest of this document. The next chapter provides background material upon which further chapters will build.

Chapter 2

Background

This research focuses on distal reward learning and behavior. Episodic memory is a class of biological neural memory and recall function that is critical to distal reward behavior. Episodic memory is the memory of event sequences, including people, places, things, etc. This chapter provides information about the anatomy, physiology, and psychology of the mammalian brain as is pertinent to episodic memory and distal reward behavior. It also discusses category theory, the Categorical Neural Semantic Theory (CNST), and constituent neural submodules that were used in design of the architectures. The chapter concludes with a description and demonstration of the temporal semantic representation scheme used in the architecture. Neural anatomy and physiology are examined in the context of episodic memory, for which the links to distal reward behavior are made. The neural submodules model neural networks that exhibit unsupervised learning behavior. A theory of temporal semantic representation provides a foundation upon which to build episodic memory utilizing neural processes.

2.1 Distal Reward

2.1.1 Characterization

Distal reward describes a class of behavior that requires associating sensory observations, motor actions, and rewards through time. Learning these associations involves reinforcement of neuronal activity due to the experienced reward [5] [6] [7] [8] [9]. Often, the reward comes some time after reward-predicting sensory observations or reward-triggering actions. This creates an explanatory conundrum known in the behavioral literature as the distal reward problem [6] and in the reinforcement learning literature as the credit assignment problem[10] [11] [12] [13] [9] [14]. Any number of sensory observations and actions might proceed the reward. In order to learn and be able to attain reward, the agent must assign credit to those preceding actions which participated in the attainment of reward. An animal's neural system must determine which out of all possible neural activation patterns are responsible for the reward. Moreover, the responsible patterns may not even be present in current neural activations when the reward arrives.

Associating observations, actions, and rewards through time differs from learning problems that involve instantaneous decisions which can be thought of as sense-react behavior. Problem solving through time requires a form of temporal reasoning. Some simple organisms sense a chemical gradient that signals food and react by exhibiting motive response up the concentration gradient. Biological examples of motive response in reaction to chemical concentration gradient, known as chemotaxis, are observed in slime mold [15] and Escherichia coli [16]. These forms of reward seeking behavior are a function only of the instantaneous changes in observation indicating the direction of the reward. Behaviors similar to chemotaxis can be observed in higher order animals. A simple experiment might train a rat that if a certain button is stepped upon when a certain tone is heard, there will occur a food reward. Or, that given a choice between square and round doors, reward will lie through the

round door. These sort of behaviors are of the simple sense-react type. For example, a subject optically (or otherwise) senses the two doors, then reacts as experience has taught will immediately result in reward given the current sensory input.

Distal reward behaviors are more complex. Solving a distal reward problem requires behavior that will not result in immediate reward, but will result in reward later. Immediate cues, at the time of action, do not tell an agent a priori what action might lead to reward. Take a door with keypad lock for instance. You might know that reward lies behind the locked door, and that some combination of key presses will disengage the lock. If we let each key press be a discrete event in time and let there be an unknown number of digits in the combination, then this can be modeled as a distal reward problem. Let the first lock combination digit be a 7. Just pressing the 7 key accomplishes nothing. Let the next and final combination digit be a 9. Again, just pressing the 9 accomplished nothing. In order to solve this unlocking problem, a subject must press 7 then 9 in temporal sequence. Temporal reasoning is inherently distinct from and more complex than simply sensing and immediately reacting.

In order to solve a distal reward problem , eposidic memories that encode temporal sequences, or some functional equivalent, is required to link actions with temporally distant consequences. The Morris Water Maze [15] task is a particular distal reward problem where the agent cannot sense the reward, but must perform a sequence of behaviors that will, over time, lead to the hidden reward.

Dopamine is a neurotransmitter strongly implicated in the formation of memories for episodes which occur in a context of reward. Nitz, Kargo, and Fleischer [17] present an experiment which links dopamine signaling with distal reward behavior. Nitz et al. uses two sample sets of biological rats, one normal and one with genetic knockout of a particular dopamine receptor type. The rats perform a distal reward maze task consisting of three left/right decisions in succession. The dopamine receptor deficient rats exhibit impaired decision making at the choice points that are

temporally distal to the reward. This result leads to consideration of dopaminergic action as an important component of distal reward behavior.

2.1.2 Other Distal Reward Models

Other published distal reward behavior models are introduced in this section. These models will be revisited in the Summary and Conclusion chapter. A number of concepts occur through several of the reviewed models as well as this dissertation research. Before continuing with discussion of the comparable models, we set context by highlighting embodiment and cortical hypercolumns, a couple of those common concepts.

One of the characteristics of experimental design seen in many of the reviewed papers is partial embodiment. For our purposes, a full embodiment would entail system input from sensory observation of an environment, processing of that input then action which alters the environment, followed by re-sensing of the now changed environment. Additionally, a fully embodied system must internalize the observation to action loop. For instance, if nodes inside the system are arranged in a two dimensional grid and activated to represent the system's location in two dimensional space, this is not a full embodiment. In order to activate the grid nodes, some external processing had to take place and determine where the system was located then push this information to the appropriate grid node. An embodied system is implicitly self-contained, therefore when external processing is required for operation of a system we do not consider that system fully embodied.

A second concept seen in several of the reviewed papers is cortical hypercolumns. A cortical hypercolumn is a group of neurons in the neocortex physically arranged in a column perpendicular to the cortical surface, where the neurons in a hypercolumn have nearly identical receptive fields. The receptive field of a neuron is the region of space in which sensory observations will create a stimulus to the neuron. Hypercolumns can be subdivided into minicolumns, where neurons within a minicolumn

encode similar features. Minicolumns subdivide the feature space to which a given hypercolumn is sensitive. For instance, consider hypercolumns within visual cortex which responds to stimuli within a small portion of the visual field. Minicolumns of those hypercolumns respond to particular line orientations within the particular portion of the visual field [18].

The other distal reward models are described in a common form. First the model is introduced. Then, the treatment of embodiment is discussed. Then, a summary description of the model is provided as adapted from published articles. Finally, model operation and experimental results are described, again as adapted from the authors' publications.

2.1.2.1 Izhikevich

Izhikevich [19] presents a spike-timing-dependent plasticity based model. Equations 2.1 and 2.2 describe the learning model for synaptic weights. The state of each synapse is modeled using two phenomenological variables: synaptic strength/weight, s, and activation of an enzyme important for plasticity, c. Extracellular concentration of the neurotransmitter dopamine is represented by d and $d(t)$ is the Dirac delta function that step-increases the variable c. Firings of pre- and postsynaptic neurons, occurring at times $t_{pre/post}$, respectively, change c by the amount $STDP(\tau)$. $\tau = t_{post} - t_{pre}$ is the interspike interval. τ decays to $c = 0$ exponentially with the time constant $\tau_c = 1s$. The decay rate controls the sensitivity of plasticity to delayed reward. The following is a more detailed description of the Izhikevich model as adapted from his 2007 paper [19].

$$\dot{c} = -c/\tau_c + STDP(\tau)\delta(t - t_{pre/post}) \tag{2.1}$$

$$\dot{s} = cd \tag{2.2}$$

2.1.2.1.1 Embodiment Izhikevich's model is discussed here because it is computational and exhibits distal reward behavior in experimentation. However, Izhikevich's experimentation is in no way embodied.

2.1.2.1.2 Description Izhikevich's model operates at the neuronal spike timing level. This neuron spike level is two levels of detail closer to a molecular model than our research. One level of detail spans from neural ensembles such as ART down to individual neurons where action potential frequency is represented by activation level. Action potentials being the actual chemical-electrical signal which propagate through biological neurons. The next level of detail spans from activation level representing action potential frequency down to the timing of firing of individual action potentials. The timing of individual action potentials becomes important in the Izhikevich model because of spike timing dependent plasticity (STDP) which is the underlying learning mechanism by which the model operates. Note the distal reward learning is the learning behavior exhibited. STDP is the learning mechanism used to enable that learning behavior.

Given the inter-neuron dynamic described by equations 2.1 and 2.2, Izhikevich's simulations are carried out using networks that consist of 1000 randomly connected spiking neuron models [20]. Of the networks' 1000 neurons, 800 are excitatory neurons with nominal spiking speed and 200 are inhibitory neurons with fast spiking speed [21], representing layers 2 and 3 of a cortical minicolumn. Neurons are randomly connected with 10% probability so that there are an average of 100 synapses per neuron.

2.1.2.1.3 Operation and Results With the above described network, Izhikevich performs four experiments: Reinforcing a Synapse, Classical Conditioning, Stimulus-Response Instrumental Conditioning, and Shift of Dopamine Response from Non-Reward-Predicting Stimuli to Reward-Predicting Stimuli in Classical Conditioning. For the first experiment, Reinforcing a Synapse a synapse that connects two

excitatory neurons is randomly chosen and that synapse's synaptic strength is manually set to zero ($s = 0$). The firing rate in the network is around 1 Hz, so every few minutes the postsynaptic neuron fires by chance within 10 ms after the presynaptic neuron. Every time such an event occurs a reward is delivered to the network in the form of a spike of extracellular dopamine with a random delay between one and three seconds. This experiment could be interpreted as the simplest form of instrumental conditioning [9].

The second experiment, Classical(Pavlovian) Conditioning, consists of rewarding a reward-predicting stimuli embedded into a continuous stream of a large number of irrelevant but equally salient stimuli. To simulate the experiment, 100 random sets ($S1, S2, ..., S100$) of 50 neurons each are chosen to represent 100 random stimuli. To deliver a stimulus, say $S1$, all 50 neurons in the set $S1$ are stimulated with a 1-ms pulse of superthreshold current. Next, a continuous input stream is formed consisting of stimuli S_k ($1 < k < 100$) in the random order with a random interstimulus intervals between 100 ms and 300 ms. S1 is treated as the reward predicting stimuli and the other stimuli as distracters. For every occurrence of $S1$, a reward is delivered in the form of the increase of extracellular dopamine with a random delay of up to 1 second. The delay is large enough to allow many neurons in the network to fire a spike and to allow a few irrelevant stimuli during the waiting period. Thus, the network receives reward on average every 20 s caused by an unknown (to the network) firing pattern embedded in a sea of random spikes and distractors.

In the third experiment, Stimulus-Response Instrumental Conditioning a typical instrumental conditioning experiment is simulated. A network of 1000 cortical spiking neurons are reinforced to produce an appropriate motor response to a stimulus. First, a random group of 50 neurons are chosen, called S, that represents the input stimulus to the network. Two random nonoverlapping groups of 50 excitatory neurons each are chosen, called A and B, that give rise to 2 motor responses of the network. To deliver the stimulus, a strong 1-ms pulse of current is injected into the

neurons in S to make them fire. Their coincident firing typically evokes a few spikes in the other neurons in the network. During a 20-ms time window after the stimulation, the number of spikes fired by neurons in A and B are counted, denoted as $|A|$ and $|B|$, respectively. The network is considered to exhibit response A when $|A| > |B|$, response B when $|B| > |A|$, and no response otherwise (e.g., when $|B| = |A| = 0$). One might think of neurons in groups A and B as projecting to 2 motor areas that innervate 2 antagonistic muscles; to produce a noticeable movement, one group has to fire more spikes than the other group.

Izhikevich's final experiment, Shift of Dopamine Response from Non-Reward-Predicting Stimuli to Reward-Predicting Stimuli in Classical Conditioning, reproduces the basic phenomenology of shifting the release of dopamine in response to non-reward-predicting stimuli to an earlier reward-predicting stimulus. The details of this experiment can be read in the *Solving the Distal Reward Problem through Linkage of STDP and Dopamine Signaling* paper [19]. Considering results from the above described experiments. Izhikevich's conclusion is that modeled form of dopamine modulation of STDP provides an elegant solution to the distal reward/-credit assignment problem.

2.1.2.2 Samsonovich and Ascoli

The Samsonovich and Ascoli [22] model reproduces spatial pathfinding, nonspatial problem solving, and episodic memory retrieval behaviors using a computational hippocampal model.

2.1.2.2.1 Embodiment In the context of their spatial task, the Samsonovich and Ascoli model is not fully embodied. Rather, internal model nodes are directly set to a unique activation pattern for each small neighborhood in the environment. In order for internal nodes to be repeatably set based on external location, sensors and processing must exist and occur external to the agent. A full embodiment contains all

environmental sensors, information processing, and resulting agent actuation within the agent itself.

2.1.2.2.2 Description Just as do the computational hippocampus models of this dissertation research, the Samsonovich model makes explicit use of θ rhythm phase changes observed in biological hippocampal neuron activation [23] [24]. Samsonovich et al. theorize that the function of phase changes is to explore possible directions of motion, where as this dissertation research theorizes that theta rhythm phase change is the mechanism which allows for alternation between bottom-up and top-down processing. While theta rhythm plays an explicit role in the Samsonovich and Ascoli model, it is only an implicit enabling mechanism in the dissertation model. The following is a more detailed description of the Samsonovich and Ascoli model adapted from their 2005 paper [22].

2.1.2.2.3 Operation Samsonovich and Ascoli presuppose that, during exploration, the environment is navigated randomly, but extensively. Each visit to a particular location results in a strong activation of the corresponding CA3 and CA1 model units. At various arbitrary moments during exploration, corresponding to the occurrence of sharp waves, the system pauses, and the current location is taken as a potential future goal. At this point, the activity mode changes with reactivation of recently active CA3 place cells, with a firing rate proportional to the recency of their last strong activation. In addition, CA1 cells representing the current location are also reactivated (Buzsaki 1989). As a result, CA3 place cells become associated with the selected CA1 cell whose place field represents a potential future goal. The strength of the associations in this model is proportional to the recency of a place-cell firing during exploration as described by Equation 2.3.

$$W_{ij}^{new} = max\{W_{ij}^{old}, \frac{1}{t - t_j}\} \tag{2.3}$$

The efficacy (weight) of the synaptic connection from CA3 unit j to CA1 unit i is W_{ij}, t is the current moment of time, and t_j is the time stamp of the last visit to the location associated with the unit j during the running session. Initial conditions for weight vectors Ws are arbitrary small values below $1/L_{max}$, where L_{max} is the maximal (allowed) trajectory length that can be associated with the goal during the training session. Reciprocal connections weights are set to a constant ($W_{ii} = 2$). At the conclusion of an associative learning event, the system resumes random exploration of the environment until another arbitrary potential goal is selected. Then the associative process repeats.

Through random exploration until a preselected goal is reached, the CA3 model units acquire weighted connections to the CA1 representation of the goal. At this point, the network of place cells encode information sufficient for finding a short path to this goal location, starting from any location within the environment. This is accomplished by a simple algorithm; at each location, several randomly generated possible local moves are generated, and the move that produces the strongest excitation of the place cell associated with the goal is selected and performed. This process is repeated until the goal is reached.

2.1.2.2.4 Results Samsonovich and Ascoli show successful navigation results with their model. An average ratio of 60:1 for the length of the path leading to goal before and after learning is provided as a summary statistic of navigation. Their model is also shown to perform a non-navigation task that demonstrates a memory indexing function of the same model which was shown to provide navigation function. They state that their results "provide a substantial ground for a theoretical unification of the two best known hippocampal functions, suggesting that (1) the pathfinding function of the hippocampus, in addition to its associative and memory indexing functions, may be vital for retrieval of certain episodic memories, and (2) the hippocampal spatial navigation function could be a precursor of its memory

function." [22]

2.1.2.3 McKinstry et al.

McKinstry et al. [1] model neural behavior that accounts for sensory integration over time. Their methods do not model episodic memory, nor hippocampal function, but their model is comparable to this dissertation research in that it addresses temporal requirements to enable embodied performance.

2.1.2.3.1 Embodiment McKinstry et al. perform a physical embodiment of their model with the Darwin IX mobile robotic platform. Positive aspects of a physical embodiment include sensory observations and environmental feedback to actions which occurs in a world with exactly the same physical laws in which biological neural systems are embodied. For instance, changes in perspective due to movement are not simulated to float precision in a microprocessor, rather in a physical embodiment perspective changes are a natural consequence of location change. Negative aspects of a physical embodiment include the potential for numerous confounding factors. For instance, some qualities of motor feedback may be due to lose tolerances in a gearbox. It may be interesting that a given neural system can accommodate motor artifacts due to gear tolerance. However, whatever adaptations occur to make that accommodation, they are not reflecting neurophysically inspired behavior. That is, theoretical insights from experimental results can be confounded by mundane physical effects. The following is a more detailed description of the McKinstry model as adapted from their 2008 paper [1].

The embodied Darwin IX agent was a circular wheeled mobile robotic platform with a vision system and circumferential distance sensors. The Darwin IX was augmented by a whisker array on each side. The whisker array supplied input to the simulated nervous system. Also, the whisker array supported innate avoidance and wall-following behaviors.

2.1.2.3.2 Description Darwin IXs default behavior was to move forward at a speed of about 8 cm/s in a straight line. Darwin IX also had an innate wall-following capability based on signals from the whisker array. The whisker array consisted of a column component and a row component. An innate freezing/avoidance response was triggered in Darwin IX upon detection of a simulated foot-shock by a downward pointing infra-red sensor that measured changes in reflectivity of floor surface. Construction paper of the same color as the floor but more reflective, was placed upon the floor in locations to trigger this innate avoidance response. This freezing/avoidance response consisted of stopping for about 4 s followed by a turn away from the foot-shock simulating stimuli. The signal from the infra-red sensor resulted in a neuromodulatory alteration of synaptic strength in the simulated nervous system.

Darwin IX's neurons were simulated by a mean firing rate model. Each neuronal unit represented the average activity of approximately 100 neurons over 100 ms. Seventeen neural areas, 1101 neuronal units, and about 8400 synaptic connections made up the Darwin IXs simulated nervous system. Each simulation cycle took roughly 100 ms of real time. During each simulation cycle every neuronal unit and plastic synapse was updated once. Darwin IX contained neural areas analogous to the somatosensory pathway in the rat brain, specifically the ventromedial nuclei of the thalamus, and primary and secondary somatosensory areas. By the modeled connectivity, a deflection of a particular sequence of Darwin IXs whiskers would create a spatiotemporal pattern of simulated neural activity. Such a dynamic sequence was comparable to that observed in the rat brain [25].

2.1.2.3.3 Operation In the McKinstry et al. experiments, two models were described that respond to real-world stimuli by maintaining sensory observation signals across temporal delays using two different mechanisms modeled on biological nervous systems. One strategy was through delays in neuronal responses such that spatiotemporal stimuli were mapped into activity patterns. The delayed neural re-

sponse strategy converted artificial whisker signals into patterns of activity that facilitated texture discrimination. The other model strategy was to delay synaptic plasticity such that a sensory observation was associated with a temporally previous signal. The delayed plasticity strategy allowed visuomotor stimuli to predict future collisions and resulted in preemptive motor actions.

2.1.2.3.4 Results After learning, Darwin IX's visual responses predicted future collisions and enabled the embodied agent to exhibit smooth movement down the middle of the pathways marked by cones. The three different test courses had turns of varying degree. Successful navigation required a combination of braking and turning of the proper magnitude at the proper time, see Figure 2.1. The delay eligibility trace learning rule was successful across all three course with a four second delay. Longer and shorter delays were used in experimentation, but did not perform well.

McKinstry et al. observe that biological systems cannot afford to store all signals indefinitely. Thus a recurring theme in neuroscience is how the nervous system maintains and integrates relevant signals across different timescales. In their paper, they described two real world embodied models that maintain observed input across temporal delays using two different mechanisms modeled on mechanisms found in biological nervous system. They show experimentally that the embodied model of these biological systems exhibits successful behavior.

2.1.2.4 Dolle et al.

Dolle et al. [26] investigated competition between parallel memory systems through modeling of different navigation strategies. The Dolle experiments used a simulated Morris water maze. The two modeled memory systems were labeled "taxon expert" and "planning expert." For the taxon expert, a form of sensory input was simulated in that for a population of sensory cells, activation of a particular cell indicated the presence of the platform in a particular direction.

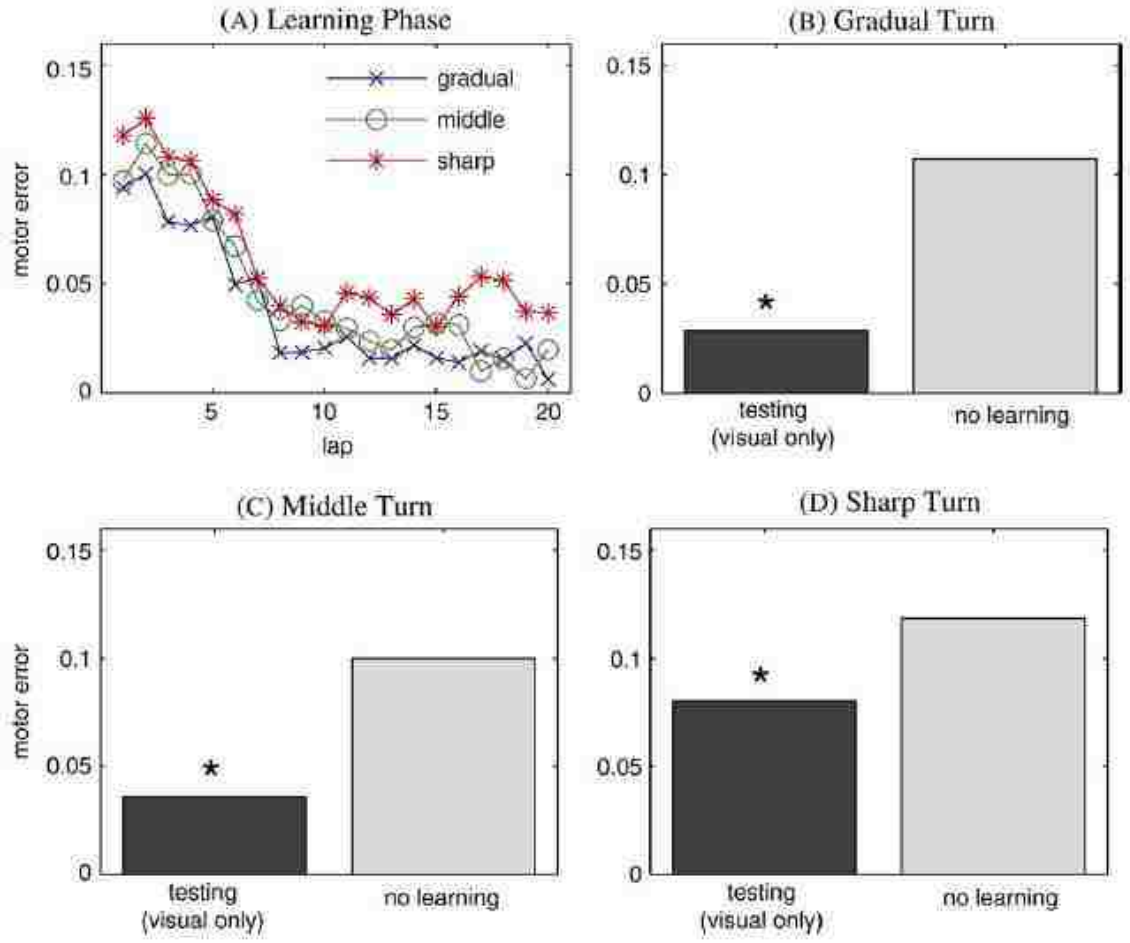

Figure 2.1: McKinstry results figure [1]. Training and testing the device on different curved courses. The mean motor error from five subjects is shown in the plots. A. Learning curves during training on the gradual turn, sharp turn, and middle turn courses. BD. Motor errors were significantly lower in the test group, which had access to only visual cues, than in the control or "no learning" groups on the Gradual course (B), Middle course (C), and Sharp course (D). * denotes $p < 0.01$ Wilcoxon Rank Sum test.

2.1.2.4.1 Embodiment The taxon sensory input was a form of vision which could only see the platform, and which exhibited winner take all output in that only one vision node was active to indicate the closest match for platform direction. The planning expert used direct stimulation of modeled hippocampal cells, as seen in the [27] model. The planning expert memory system was not embodied in that

it received no sensory input. The taxon expert had a limited embodiment in that platform sensors provided observation of one part of the environment, and that part of the environment logically was modified due to action by the embodied system.

2.1.2.4.2 Description For the taxon navigation system, visual input was encoded in a population of 100 Sensory Cells (SC) in which an active cell i indicated the presence of the platform landmark in direction i. Motor response to the visual stimulus was encoded by 36 Action Cells (AC), so that one action coded for a direction every $2\pi/36$ radians. Learning was performed by a Q-learning algorithm adaptation. A Gaussian update rule allowed ACs near the winning cell to also update their weights.

2.1.2.4.3 Operation The planning navigation system started by learning a topological representation of the environment during pretraining sessions then used this representation to place the goal location within that representation and plan an optimal path towards it during the goal planning phase. The map took as input the activity of a population of place cells. This population was provided by a hippocampus model simulating the enthorinal cortex (EC) and the dentate gyrus (DG). EC modeling was based on grid cells, which are receptive to specific spatial orientation. The grid cells activation were appended to a vector of 100 neural modeling units that represented visual inputs from landmarks outside the maze. EC cells were then fed to the DG, which trained place cells by means of Hebbian learning. Then sparse activation was obtained through use of a filter function that only kept a few cells active and set the others to zero. The planning system then normalized final DG activation and processed it to build the nodes of the planning graph. For the planning graph, a pool of 100 nodes was connected to the DG at random synaptic weights, and the same sparse learning used in the DG was replicated during the map building phase. The firing rate of a PG node j was computed as in Equation 2.4.

$$r_j^P = f_j(\sum iW_{ij}^{DG,PG} DG_i, s^{PG}) \tag{2.4}$$

$W_{ij}^{DG,PG}$ is the synaptic weight linking the DG place cell i to the planning graph node j and $f_j(x, s^{PG})$ is the same non-linear function as in the DG, returning a sparse encoding of x, with a sparseness level of s^{PG}. The synaptic weights were learned following a Hebbian rule described by Equation 2.5.

$$\Delta W_{i,j}^{DG,PG} = \alpha^{DG,PG} r_j^P (DG_i - W_{i,j}^{DG,PG}) \tag{2.5}$$

The direction of movement required to move from the location encoded by node i to the location encoded by node j was stored as the link weight between the nodes. Goal planning began when the goal position was found, at that time the closest node was set to a reward value. Then, using the planning graph, an optimal path to the goal was determined by a bio-inspired activation-diffusion mechanism based on Dijkstras algorithm for finding the shortest path between two nodes in a graph.

2.1.2.4.4 Results A theory of parallel control systems in the rat brain was modeled in the Dolle et al. work. The existed preceding computational models of navigation that rely on similar learning methods such as reinforcement learning for different navigation strategies [28] [29] [30]. In these previous models, strategy selection was usually carried out through a non-adaptive mechanism [31] [32]. The implementation of adaptive selection among different learning modes (including topological representations) was a characteristic feature of the Dolle et al. research.

Dolle et al. experimentally demonstrated the ability of their model to select navigation strategies in two simulation experiments. They note that interaction among several spatial memory systems may improve the performance of animals. This improvement may come through speeding learning by cooperation among different strategies, or competitive processes that prevent the application of sub-optimal

strategies. The also note that design of robots for a wide range of behavioral tasks may be enabled by better understanding these interactions through computational modeling.

2.1.2.5 Weber and Triesch

Weber and Triesch [33] developed a reinforcement learning based model for goal directed feature learning. A characteristic of their model was that it received sensory input which was modified by model actions.

2.1.2.5.1 Embodiment The Weber and Triesch embodiment was limited in a different way that the Dolle et al. model. While the Dolle embodiment was limited by only sensing one element in the environment, the Weber and Treisch model operated with an overhead perspective in a non-real world space, as well as non-deterministic simulated macro level physics in the environment. The environment consisted of several two pixel length bars on a two dimensional surface. When the model system took a movement action in a given direction, the sensory observation of the blocks translated appropriately in space. The characteristic of the environment space which was non-real world was that when a bar left one side of the agent's perception due to embodied action, it entered the opposite side. This was an embodiment with sensory input equivalent to an agent navigating on a torus while looking down at the "floor" through a tube. The non-deterministic portion of the simulated environment physics was that for any given embodied movement, there was a 20% chance that each bar would fail to translate in response to the movement. So, the elements of the environment were usually modified according to the agent's actions, but not always. Out of several bars in the environment, one particular bar was the reward feature. Reward was given when the reward feature was located in a certain percieved location.

2.1.2.5.2 Description The Weber and Triesch model consisted of input, hidden, and output layers. The input layer recieved a sensory vector I. The hidden layer learned features of the input and encoded a winner-take-all state vector s, i.e. only one hidden layer unit was active at a time. The output layer held winner-take-all vector a, that represented the currently selected action. The model's neural network was feedforward with full connectivity between adjacent layers.

2.1.2.5.3 Operation Each trial lasted until reward was found, which resulted in $r = 1$, with $r = 0$ until reward was found. The agent was placed at a new random position at the beginning of each trial. The learning algorithm was similar to SARSA [13]. In step 0, an action was performed and new sensor values I were incident upon the input layer. In step 1, the hidden layer activations updated according to the feature weights W between the input and hidden layer along with the winner-take-all mechanism. In steps 2 and 3, an action was selected using the weights Q between the hidden and output layer. Step 4 computed the value v' of the new state based on the next possible actions and the potential weights Q that would be used corresponding to each next possible action. In step 5 the prediction error δ was computed between the time-discounted current value and the earlier value v, the reward r was also considered. In step 6, the action layer weights Q updated through a δ-modulated Hebbian rule with state s and action a as pre- and post-synaptic values. The feature layer weights were also updated by a δ-modulated Hebbian rule, with pre-synaptic activations I and post-synaptic activations s.

2.1.2.5.4 Results Weber and Triesch's results show the learning progress of their reinforcement learning based system. An actor behaving at random averages 50 steps before finding the reward. After 100,000 iterations the network finds reward in five steps on average.

2.1.2.6 Martinet et al.

Martinet et al. [27] described a model which simulated physical maze traversal. The Martinet model was implemented in a simulated robotic platform. The neural model employed was based on cortical hypercolumns and minicolumns.

2.1.2.6.1 Embodiment Embodiment is limited in that the model receives direct neural stimulation from external environmental analysis as opposed to relying solely on sensory input.

2.1.2.6.2 Description and Operation In the Martinet et al. model, the neural system was organized into hypercolumns and the hypercolumns were organized in minicolumns. An unsupervised learning scheme made the activity of each minicolumn selective to a specific state-action pair $(s, a) \in S \times A$. A given action $a \in A$ represented the allocentric motion direction of the agent when it performed a movement that transitioned between two locations $s, s' \in S$. By effect of the learning algorithm, minicolumns that were selective for the same spatial location $s \in S$ were grouped to form a hypercolumn. This is a divergence from biological neural networks where minicolumns are physically located in hypercolumns. Given the described learning, hypercolumns consisted of a set of minicolumns that were incrementally recruited to encode all the state-action pairs $(s, a_{1...N}) \in S \times A$ experienced by the agent at location s. All minicolumns in a hypercolumn competed with each other during navigation planning to locally infer the most appropriate goal-directed action.

Minicolumns of the model were made up of two computational units, one represented supragranular layer (SL) neurons and the other represented infragranular layer (IL) neurons. Mean firing activity of a population of cortical neurons in cortical layers II-III, and V-VI, respectively were simulated by the activation of SL and IL nodes. Each minicolumn received three different sets of afferent connections: hippocampal inputs, collateral afferents, and motivation dependent signals. Hip-

pocampal inputs encoding spatial location connected to IL neurons through plastic connections with synaptic efficacy determined by the weight distribution w^h. Collateral afferents from adjacent cortical columns connected to SL and IL neurons. These lateral connections underwent incremental learning and were used to encode environment topology which in turn was used in the activation diffusion planning method. The final columnar input was to SL neurones which received motivation-dependent signals.

2.1.2.6.3 Results Experimentation was divided into three phases with different maze configurations used for each phase. Initially, the agents planned maze navigation in the absense of both block A and B. The results showed that the model learned to select the shortest goal-directed pathway P1 significantly more frequently than the alternative trajectories P2, P3 (ANOVA, $F_{2,297} = 168.249, p < 0.0001$).

During the next training phase (consisting of 156 trials), a block was introduced at location A, which forced the agents to update their topological maps dynamically, and to plan a detour to the goal. Results reported for equivalent biological experiments provide strong evidence of a preference for the shortest detour path P2. Correspondingly, in the Marinet et. al simulations (Fig. 3B) a significantly larger number of transits through P2 compared to P3 were observed (ANOVA, $F_{1,198} = 383.068, p < 0.0001$).

In the last training phase, block A was removed a new block added at location B. This phase was designed to test the "insight" hypothesis. The "insight" hypothesis suggests that after an agent learns path P1 and later encounters unexpected block B, the agent will go through the longer but effective path P3 as opposed to the apparently shorter but now blocked path P3. In biological results from the same experiment using rats [34], Tolman and Honzik concluded that rats were able to inhibit the previously learned behavior of selection path P2 after a failure of path P1. Similar to rats, the Martinet et al. agents exhibited a statiscally significant

preference for P3 compared to P2 (ANOVA, $F_{1,198} = 130.15, p < 0.0001$).

Finally, Marinet et al. compared the action selection policy of learning agents with the theoretical behavior of randomly behaving agents. The results indicate a significantly better performance of learning agents compared to random agents (ANOVA, $F_{1,196} = 7.4432, p < 0.01$).

2.2 Episodic Memory

As discussed in Chapter 1, episodic memory is central to the neural architecutre developed in this work to study distal reward. In mammals, the hippocampus has been shown to be a key structure in the formation of episodic memories. The utility of episodic memory in this work is to bind temporal episodes of sensory experience with episodes of proprioceptive (motor sensors) encoding of our agent's movement. This subsection reviews neuroscientific literature describing both episodic memory and the hippocampus. The review of episodic memory begins with a more psychological examination of memory to narrow down the behavioral qualities of episodic memory, and then transitions to examining evidence for neurophysiological correlates of episodic memory.

Episodic memory encodes temporal sequences of sensory information. More specifically, the neural mechanism for what we label "memory" encodes activations that can be reconstructed as temporal sequences of perceived sensory information. That is to say, neural memory is not a video record of experiences, but rather a temporal series of neural activations that upon recall is reconstructed into a "video"-like perceived memory. The term "video" is used loosely, as it is a completely subjective opinion as to how closely a particular animal's memory recall resembles video playback.

In biological organisms, memory is used to make plans for future actions. Likewise the artificial neural architecture will use memory to guide future actions. Biological

existence involves interaction between retrospective memory and prospective memory [35]. Retrospective memory is the capacity for recollection of general knowledge and history of previous actions and their outcomes. Prospective memory is the application of knowledge and history in directing ongoing decisions and behavior.. To help illuminate architectural decisions, experimental evidence on the cognitive and neural mechanisms of recollection is outlined, and then retrospective memory results from experimental studies in cognitive science is considered. As mentioned above, one of the major neurophysiological structures involved in memory is the hippocampus, see Figure 2.2, and therefore the following subsection starts the review win an examination of the hippocampus.

2.3 Neuroscience and Cognitive Psychology Background on Hippocampus

The hippocampus is a neurophysiological structure involved in memory. Experimental and observational data suggests specific cognitive and neural information processing behaviors for which hippocampus is responsible or at least necessarily involved. By necessarily involved, we mean necessary for a function, but without assuming sufficiency for that function. Much of the information known about the function of hippocampus results from lesion studies. That is, what functions does an organism loose when all or part of the hippocampus is damaged or removed. Many of the following studies involve accidental or experimental lesioning.

The following references to research about hippocampal function are ordered into subsections. The first subsection addresses evidence for hippocampal function in humans, it cites accidental lesion cases. The second subsection cites studies that refine hippocampal function by conducting behavioral experiments on animals. The third subsection focuses in on studies that involve episodic memory, in particular the temporal aspect of episodic memory.

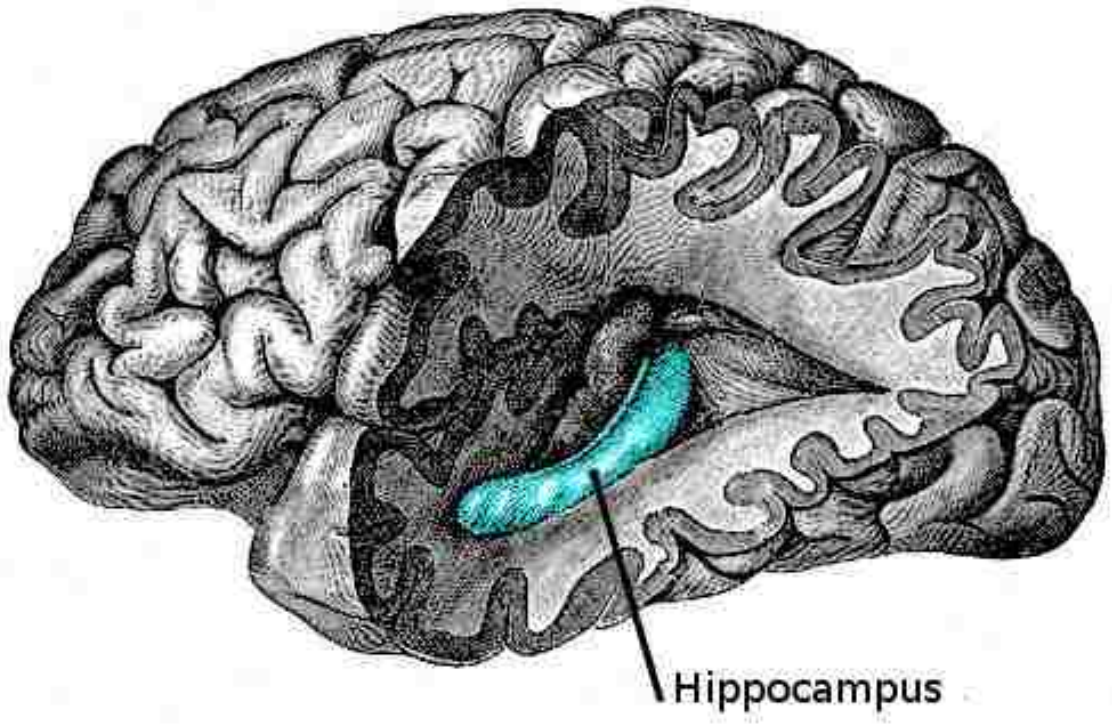

Figure 2.2: Modified from a scan of a plate of "Posterior and inferior cornua of left lateral ventricle exposed from the side" in Gray's Anatomy.

2.3.1 Human Cases

Some cases of mild hypoxia (lack of oxygen to the brain) result in damage largely confined to the hippocampus. An analysis of recognition memory performance showed that this localized hippocampal damage caused a severe decrease in recollection performance, while other memory measures remained normal [36]. Specifically, familiarity performance was not hindered by hippocampal damage. The result from this research that is most pertinent to this dissertation is that hippocampus plays a particular role in recollection.

Decreased recollection performance with retained familiarity performance was also reported in a case of meningitis causing relatively localized hippocampal atrophy

[37]. The observed involvement of hippocampus in recollection memory tasks as differentiated from familiarity memory tasks suggests that hippocampus is involved in detailed sensory memory activity, while some simple sensory familiarity matches are possible without hippocampal involvement. Hippocampus is explored further to determine what features and characteristics are important for an episodic memory model which will perform a distal reward task.

In a study on the coding properties of hippocampal neurons in humans, Ekstrom et al. [38] recorded neural activations in subjects as they played a taxi driver game, searching for passengers picked up and dropped off at various locations in a virtual reality town. They observed that many of these cells fired selectively associated with specific combinations of a place and the view of a particular scene or a particular goal. These and other studies indicate that, in rats, monkey, and humans, a prevalent property of hippocampal firing patterns involves the representation of unique associations of stimuli, their significance, specific behaviors, and the places where these events occur [39].

2.3.2 Animal Evidence for Hippocampal Function

The preceding studies involved human subjects, a much greater amount of experimental work exists using other animal subjects and is considered next. Several studies have shown that damage limited to the hippocampus results in deficits in memory formation for the context or location where items were once experienced [40]. In one recent study, rats were initially exposed to two objects in particular places in one of two environmental chambers [41]. In subsequent recognition testing, the place of the object or the context was changed. Normal rats increased their exploration of objects that were moved to new places or put in novel contexts. By contrast, rats with hippocampal damage failed to recognize objects when either the place or context was changed [42].

Several investigators have argued that animals are indeed capable of remembering

the temporal as well as spatial context in which specific stimuli are experienced [43] [44]. To further explore these aspects of episodic memory, Eichenbaum developed a task that assesses memory for events from a series of events that each involve the combination of an odor (i.e. what), the place in which it was experienced (i.e. where), and the order in which the presentations occurred (i.e. when) [45]. On each of a series of events, rats sampled an odor in a unique place along the periphery of a large open field. Then, memory for when those events occurred was tested by presenting a choice between an arbitrarily selected pair of the odor cups in their original locations. Normal rats initially employed their memory of the places of presented cups and approached the location of the earlier experience. Then they confirmed the presence of the correct odor in that location. Animals with selective hippocampal damage fail on both aspects of this task even though their memory for independent features of location and odor items was intact. These findings indicate that the hippocampus is critical for effectively combining focus, context, and temporal qualities of each experience to compose the retrieved memory.

2.3.3 Episodic Memory Experiments

Episodes are represented as sequences of sensory observations. Embodied agents live their lives through experience. The initial construction of perceived reality is a form of episodic buffer that contains a representation of the stream of sensory observations as they just occurred [46]. Tulving [47] distinguished episodic memory as organized in the temporal dimension, and contrasted this scheme with a conceptual organization of semantic memory. Tulving argued that the central organizing feature of episodic memory is that "one event precedes, co-occurs, or follows another." This is similar to Aristotle's [48] characterization of vivid recall: "Acts of recollection, as they occur in experience, are due to the fact that one thought has by nature another that succeeds it in regular order." These characterizations emphasize the temporal organization of episodic memories. The temporal semantic representation theory, which will be

presented later, is motivated by these insights. It will represent episodic memory as recency gradient encoded sequences of generalizations of neural activations. The neural activations hold semantic information because they are grounded through connection to the sensory input of an embodied system.

The order of events within human memory depends on hippocampal function. In a study using a design similar to the Tulving study, Hopkins and Kesner [49] found that patients with hypoxic brain injury involving shrinkage of the hippocampus are impaired in memory for the order of a series of 6 words, pictures, or spatial locations. These patients were, however, also impaired in recognition of the items, making difficult an unambiguous interpretation of a deficit in the order of the events independent of memory for the events. More recently, Spiers et al. [50] reported a selective deficit in order memory independent of item memory in a patient with selective hippocampal damage due to perinatal transient anoxia [51]. In this study, the patient explored a virtual reality town in which objects were received from virtual characters. Recognition of the familiar objects was intact, memory for the order in which the objects were received was severely impaired, as was memory for where the objects were received. Also, Downes et al. [52] reported that patients with medial temporal lobe damage that included bilateral hippocampal damage were impaired in memory for the order of presentation of words for which recognition of the items was equivalent. Also, evidence from the deferred imitation task, where subjects are required to remember an action sequence, indicate a critical role for the hippocampus [53] [54]. Thus, humans with hippocampal damage are impaired in memory for the order of events in unique episodes even in cases where recognition memory is intact. This provides evidence that hippocampus is necessary for neural representation of temporal order.

Studies on animals also show that the representation of memories by the hippocampus incorporates not only items that must be remembered, but also the events that precede and follow. For example, Honey et al. [55] provided a simple demon-

stration of the importance of temporal order in hippocampal processing, reporting that hippocampal lesions disrupted animals' normal orienting response when a pair of stimuli are presented in the opposite order of previous exposures. The specific role of the hippocampus in remembering the order of a series of events in unique experiences has been explored using a behavioral protocol that assesses memory for episodes composed of a unique sequence of olfactory stimuli [56] [57]. Memory for the sequential order of odor events was compared with recognition of the odors in the list independent of memory for their order. On each trial rats were presented with a series of five odors, selected randomly from a large pool of common household scents. Memory for each series was subsequently probed using a choice test where the animal was reinforced for selecting the earlier of two of the odors that had appeared in the series. In later sessions it was also tested whether the rats could identify the odors in the list independent of their order, by rewarding the selection of a novel odor against one that had appeared in the series. Normal rats performed both tasks well. Rats with hippocampal lesions could recognize items that had appeared in the series but were severely impaired in judging their sequential order.

A common observation across many different behavioral protocols is that different hippocampal neurons become activated during every event that composes each experience. I.e. a list of neurons becomes activated in sequence as a list of events occur which compose an experience. This hippocampal neuron activation behavior is observed during simple behaviors such as foraging for food [57] as well as learning related behaviors directed at relevant stimuli that have to be remembered in studies that involve classical conditioning, discrimination learning, and non-matching or matching to sample tasks to tests and a variety of maze tasks [58] [59]. In each of these paradigms, animals are repeatedly presented with specific stimuli and rewards, and execute appropriate cognitive judgments and conditioned behaviors. Corresponding to each of these regular events, many hippocampal cells show time-locked activations associated with each sequential event. Also, as described above, many of these

cells show specificities corresponding to particular combinations of stimuli, behaviors, and the spatial location of the event. Thus, hippocampal population activity can be characterized as a sequence of firings representing the step-by-step events in each behavioral episode.

This section has provided many examples in the biological literature to demonstrate hippocampal function particularly as it pertains to episodic memory. The computational neural architecture of this dissertation research was designed with guidance from biological hippocampal form based on the preceding evidence that biological hippocampus enables episodic memory based behavior. An extended literature search on experiments involving biological hippocampus is found in Appendix C.1.

2.4 Dopaminergic Subsystem

In the human brain, the mesocorticolimbic system houses the dopaminergic neurons, in which originates reward signals that synapse on many other brain areas. Dopaminergic signals are expected reward indicators. If an unexpected reward occurs, the dopaminergic neurons fire at an increased rate. If an expected reward does not happen, the dopaminergic neurons fire at a decreased rate. If no reward is expected, and none occurs, the dopaminergic neurons fire at a tetanic default rate [60]. For purposes of the neural architecture, we utilize a functional model whose input will detect the presence of a reward, finding the submersed platform for example, and whose outputs control learning and recall in other architectural components.

2.5 Experimental Results of Computational Cortical Hippocampal Model

Results of a standalone computational cortical hippocampal model are presented in appendix D.4. This model was developed along with the embodied neural architecture described in this dissertation, with some coevolution as each effort informed the other. The conclusion from the computational cortical hippocampal model results was that behavior was observed in the model which was measured to correspond with behavior published in the literature for human subjects [61]. This provides evidence that the computational cortical hippocampal model is an accurate representation of biological hippocampus. Some principles from the computational hippocampal model are present in the embodied architecture of this dissertation. Specifically, the scheme of creating a key pattern then binding that key pattern as part of the formation of episodic memories is carried over to the embodied architecture from the stand alone hippocampal model. The method by which the embodied architecture does this will be detailed later.

Favorable results provided confidence in the cortical hippocampal model and provided incentive for shared principles with the embodied model. The standalone model was only partially embodied. The standalone model has sensory input, but there existed no feedback to the environment, i.e. motor output which alters the environment.

2.6 Category Theory

Category theory is the mathematical discipline that is concerned with mathematical structure. It is used in this dissertation to assist in both architectural analysis and design. In analysis, category theory is used as a tool that suggests functional properties based on particular neural structures. It allows the definition of a framework

for the estimate of structure based concept capacity of a neural system. In design, category theory is used to modify the dynamics of constituent neural modules in the architecture to enhance system performance.

In the development of a categorical basis for this research, three investigations are performed. The first two investigations are detailed in Appendix A and the third in Appendix E. The first categorical investigation builds on the work of Healy and Caudell [62] by applying category theory to implemented artificial neural networks. Investigation one is where this research begins to examine how category theory might be useful in the analysis of neural architectures. The basic strategy is to measure categorical capacity in some simple neural networks. Using definitions for concept representation developed in [62], categorical capacity is a measure of how many distinct concepts can be encoded by a particular neural structure. For this investigation, a distinct concept is defined as a unique structure of neurally represented categorical objects that represent concept categorical objects. To this end, commutative diagrams are measured as they are a basic unit of structure in category theory.

The second investigation builds upon the first by developing a theoretical framework for categorical concept capacity. Whereas the applied artificial neural network investigation was limited to small neural networks, the theoretical investigation is extended to any size network of the topology considered. Commutative diagrams are again the basic unit of representing a distinct neural encoding of information from the environment.

In investigation three, categorical limits are investigated as an improvement to ART. This is advantageous to the embodied architecture in that ART is used as a module of self-organizing neural sub-structure modeling biological cortical hypercolumns. An improvement in the performance of ART can be reasonably expected to improve the behavior based performance of an embodied neural architecture using ART.

As a design tool, category theory is used to modify the dynamics of constituent

neural modules in the architecture. Category theory is used in a similar manner as calculus might be used to treat physical systems. Just as calculus would be useful in developing new forms for jet airplanes, category theory has proven useful in the design of neural structure. Calculus is useful in designing a jet, but it does not of itself tell one how a jet should be made. However, with a proper framework linking the math of calculus to the physical world, one could verify that certain designs are capable of certain tasks. E.g., it could be verified that a certain wing design would provide a certain amount of lift. Furthermore, the wing design might be usefully constrained from the outset by proper calculus based considerations. Just as with other forms of mathematics, there is generally more than one valid way to make a mathematical representation with category theory. In calculus, the volume of a cylinder can be calculated as a series of discs integrated along the length of the cylinder, or as a series of rectangles integrated through the width of the cylinder, or as a series of tubes integrated through the radius of the cylinder. Any of those mathematical representations of the volume of a cylinder are correct (given appropriate parameters for a certain cylinder). There is not "one right way" to mathematically represent the volume of a cylinder. Likewise, there is not "one right way" to mathematically represent a series of events as observed through sensors. As long as a representation is valid within the mathematic framework being employed, it is a correct representation. A contribution of this research is the empirical demonstration of a method for informing neural design with category theory such that category theory has descriptive power for the neural design. An introduction to category theory can be found in Appendix A.

As mentioned earlier, the computational neural architecture of this dissertation used ART modules as functional building blocks analogous to pattern classifying biological neural submodules such as cortical hypercolumns. The next section describes ART, a real valued variation of ART called fuzzy ART, and a supervised learning version called LAPART.

2.7 ART, fuzzy ART, and fuzzy LAPART

ART is a well established self-organizing neural network for classifying input activation patterns. The interested reader can find a wealth of literature on ART details, performance, neurophysical plausibility, and stability [63] [64] [65] [66] [67]. The following provides a brief description of ART, then introduces Fuzzy ART, LAPART – a combination of two ART systems, and finally Fuzzy LAPART. These networks play a critical role in the embodied architecture developed in this work.

2.7.1 ART

ART1 is an unsupervised learning system that operates on binary input vectors. The basic components of ART1 are an $F0$ input layer, an $F1$ internal layer, an $F2$ output layer, a vigilance node, and synaptic weight sets between each $F2$ node and all $F1$ nodes. An ART system is illustrated in Figure 2.3, label with M $F1$ nodes and N $F2$ nodes. A single $F2$ output node will classify a given input. At any given time, there is only one active $F2$ output node due to a winner-take-all lateral inhibition network between all $F2$ nodes. Each $F2$ output node is fully connected to all $F1$ nodes, where $F1$ nodes have one-to-one connections with the $F0$ input nodes. Activations propagated from the input to the $F1$ layer are unmodified except for normalization if necessary. For an ART1 implementation where inputs are already normalized, the $F1$ layer gets a copy of the input. The connections between all $F1$ nodes to each $F2$ node are weighted and represent the categorization templates which encode a generalization of all inputs that a given $F2$ output node will classify.

$$w_j(n+1) = \beta(I \cap w_j(n)) + (1-\beta)w_j(n) \tag{2.6}$$

$$\frac{|I \cap w_j|}{|I|} \geq \rho \tag{2.7}$$

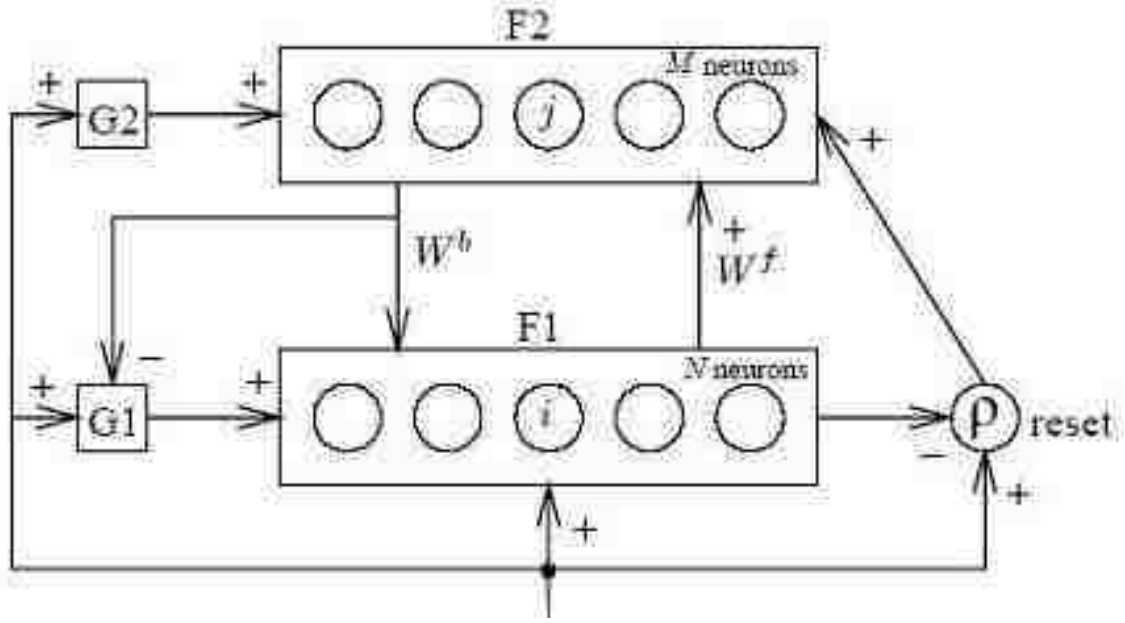

Figure 2.3: ART neural architecture

$$\frac{|I \cap w_j|}{|\alpha + w_j|} \tag{2.8}$$

Equation 2.6 shows how ART1 updates each weighted connection w_j between $F1$ and $F2$. $\cap$ is the logical AND operator. This update performs learning of classification templates. ART1 applies Equation 2.6 only to the connections incident upon the one winning $F2$ node. That $F2$ node is a single winner-take-all selection via the ART1 choice algorithm which selects the node that maximizes Equation 2.8, where I is the input vector as present in the $F1$ layer of nodes, w_j is the weight vector which comprises the classification template linking the j^{th} $F2$ node to each $F1$ node, and α is generally set to a very small positive value such as 1^{-5}. Equation 2.7 is also applied, where ρ is the vigilance parameter which controls how much difference there can be between a winning $F2$ node template and the input I. A new $F2$ node is added when no template T for an existing $F2$ node satisfies Equation 2.7. New templates, which accompany new nodes, have weighted connection values of 1. Since the new template is automatically the winning template, when no existing template

meets vigilance for a given input, the new template of all ones will instantly be eroded part or all the way to the input vector, depending upon the value of the learning parameter β in Equation 2.6. $\beta = 1.0$ is considered "fast learning" and results in the new template being updated to match the current input vector. Should that template win for a subsequent input vector, the weights would erode for any weight vector elements which are greater in value than the corresponding input vector element. Note that $w_j(n+1)$ indicates the template weight vector for the next timestep, and $w_j(n)$ is the template weight vector for the current timestep.

A useful analytical feature of ART is that, given compliment coded inputs, pattern classes are encoded as rectangular hyperboxes of dimensionality equal to half the dimensionality of the complement coded input. As each $F2$ output node is recruited, due to no template link to an already recruited $F2$ node meeting the vigilance test, the template weights with fast learning will be adjusted to a point hyperbox at the point of the current input vector. As subsequent inputs resonate with that same template, the hyperbox erodes outward and its perimeter grows. The vigilance parameter restricts how large the hyperbox perimeter may become.

As a consequence of the operation of the vigilance parameter ρ, lower vigilance results is a more general classifier and higher vigilance results in a more specific classifier. In terms of the complement coded input hyperbox visualization, a lower vigilance results in larger hyperboxes. Therefore a lower vigilance results is each class encompassing a larger region of input space. Conversely, a higher vigilance results in smaller hyperboxes which each cover a smaller region of input space. It can also be observed that for a fixed input set, lower vigilance will generally result in fewer templates formed while higher vigilance will result in increased template proliferation.

2.7.2 Fuzzy ART

Fuzzy ART is a self-organizing neural network that autonomously creates codes for clusters of real valued input patterns [64]. See Figure 2.4. For this network, pattern components are normalized to the $[0, 1]$ interval. The $F2$ layer is a winner-takes-all layer producing a sparse coded output for the best matching cluster. The granularity of the clustering is controlled by a vigilance factor ρ bounded to $(0, 1]$ that controls the level of generalization performed by this network. If ρ is close to unity, the number of $F2$ layer clusters (nodes) rapidly increases during training, essentially memorizing each individual input pattern. Conversely, if its value is near zero the number of $F2$ clusters increases very slowly during training, creating a system that greatly generalizes the input patterns, lumping many input patterns into a small number of clusters. Each cluster's prototype is encoded internally as a real valued synaptic weight pattern referred to as a template, signified as T_k in the figure. Learning in Fuzzy ART uses a specialized form of the Hebbian learning rule [68] to adapt the winning $F2$ node's template in either in a fast and slow learning modes. The former is used in this study.

The "Input System" shown in the figure is responsible for pattern preprocessing. For this study, the Input System complement codes the input pattern. Complement coding utilizes an on-response (stimulus) and off-response (one minus stimulus) encoding. For example, if $s = (s1, s2)$ is a stimulus pattern, then its complement encoding is $(s1, s2, 1 - s1, 1 - s2)$. The first two pattern components are the on-response while the second two are the off-response. If the dimensionality of the original stimulus s is d, the complement encoded input pattern has dimensionality $2d$. The use of complement coding allows the template weight patterns to be interpreted geometrically as d-dimensional hyperboxes, and improves weight convergence during learning.

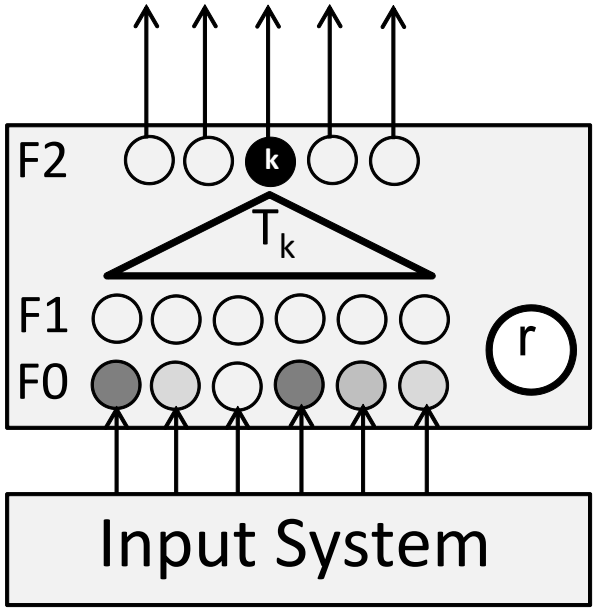

Figure 2.4: Fuzzy ART subnetwork, including an Input System responsible for complement coding of the input patterns. The network includes three layers, $F0$, $F1$, and $F2$, the latter being winner-takes-all. The neural circuitry labeled ρ controls the granularity of pattern clustering during learning. A top-down template T_k is illustrated as an open triangle for winning $F2$ node k.

$$w_j(n+1) = \beta(I \wedge w_j(n)) + (1-\beta)w_j(n) \tag{2.9}$$

$$\frac{|I \wedge w|}{|I|} \geq \rho \tag{2.10}$$

$$\frac{|I \cap w_j|}{|\alpha + w_j|} \tag{2.11}$$

Equations 2.9 2.10 2.11 describe Fuzzy ART, where variable designations are as previously given for ART1. To enable handling of real valued inputs, fuzzy ART uses the fuzzy AND operator $\wedge$ instead of the logical AND operator $\cap$ as seen in ART1.

A strength of Fuzzy ART, and ART1 is guaranteed stable category learning. If the input vector is complement coded, that is for each input element i there exists an

input element j where $j = 1 - i$, then one-pass learning will result such that no new output nodes are recruited or addition learning occurs on subsequent presentations of any input [65]. %

2.7.3 LAPART

LAPART is a supervised learning technique that makes use of ART [69] [70]. LAPART consists of a laterally coupled pair of ART1 neural networks. As a supervised learning technique, LAPART learns to associate pairs of input activation patterns. Associations are stored as connection weights between $F2$ output layers of the two ART modules which comprise a LAPART. In LAPART, the individual ARTs operate according to standard ART rules [63], and their association is governed by a logical design principle from formal semantic analysis performed by Healy [71] [72].

The coupling of ART1 systems is accomplished according to two basic rules. The first rule; an active $F2$ node in one ART1 system will trigger a rule inferring an active $F2$ node in the other ART1. The second rule; the other ART1 network may veto the prediction if it does not match existing learned data.

An improved version of LAPART, known as LAPART2 [70], is used in the artificial neural architectures of this research. LAPART2 operates identically to LAPART1 except for in the case of an input pair which pairing is not represented by the existing A $\rightarrow$ B connections between $F2$ layer nodes. In other words, LAPART2 modifies the behavior of LAPART when an incorrect prediction of ART-B is made based on the input to ART-A. With either form of LAPART, this violation of the A $\rightarrow$ B prediction prompts a lateral reset. The behavior of LAPART2 is that in the case of a lateral reset, a new $F2$ node is recruited in ART-A to encode the current input. Then, the appropriate $F2$ node is activated in the usual fashion on ART-B and a, now correct, inference connection if formed between the new node in ART-A and the correct (possibly new) node in ART-B. This LAPART2 lateral reset behavior, along with the existing LAPART operation, results in a proven 2-pass supervised

learning system [70].

2.7.4 Fuzzy LAPART

The next component to discuss is LAPART that is used by the larger neural architecture to associate temporal sequences with visual sensory patterns. The LAPART-1 network architecture is based upon the lateral coupling of two ART-1 subnetworks, referred to as A and B (Figure 2.5), which perform unsupervised classification of binary input patterns [2]. The interconnections between the two subnetworks, labeled M in the figure, force an interaction of the respective classifications performed by the ART-1 subnetworks on their inputs. This modifies their unsupervised learning properties to allow the learning of inference relationships or associations between the learned pattern classes representing their input domains. This can be thought of as supervised learning, or supervised classification. Fuzzy LAPART replaces the ART-1's with Fuzzy ART's. The two vigilance parameters A-ρ and B-ρ control the granularity of these associations.

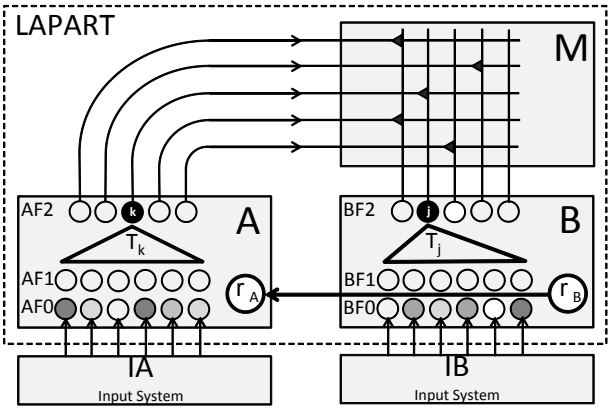

Figure 2.5: Fuzzy LAPART Subnetwork. The small solid black triangles in the M associative weight matrix represent non-zero synaptic weights learned through a Hebbian process referred to as lateral priming connections. The horizontal connection from B to A resets the A-side ART if the B-side ART rejects a postulated association. See [2] for a complete discuss of this network.

2.8 Temporal Episodes

As mentioned previously, the functionality of episodic memory in used in this study of distal reward learning. This section builds the foundation for the temporal episode encoding method used in the distal reward architecture. A practical introduction is given for the implemented technique. Then, a literature search provides context. Then, the encoding of temporal episodes in artificial neural architectures is treated in more detail and illustrated with an example.

2.8.1 Temporal Integrator

The temporal integrator TI subnetwork shown in Figure 2.6 is a layer of temporal integration nodes. It assumes sparse input codes, that is, at most one of its nodes will have unit level input at any given time. When a node in the TI subnetwork receives a unit level signal, it rapidly integrates to its maximum value in one time step. When a TI node receives a zero level input, its output decays with an exponential time constant. When its output reaches a minimum noise threshold level, it drops to zero. Each TI neuron may be thought of as an RC circuit with different rise and fall time constants. As the result of a temporal sequence of unitary sparse inputs, a unique recency gradient of activations will form across the TI layer, coding the order of input firing - the most recent input will have the largest TI layer activation, with the time reverse order having decreasing values. The TI network can be reset to set all integrator nodes to zero.

2.8.2 Temporal Sequence Learning

The temporal sequence learning network is constructed from two Fuzzy ARTs (S & L) and one Temporal Integrator (TI) subnetwork as illustrated in Figure 2.7 [73]. The dynamics of this network are as follows. Normalized real-valued input patterns are introduced to the input of the S Fuzzy ART to simulate a temporal sequence of

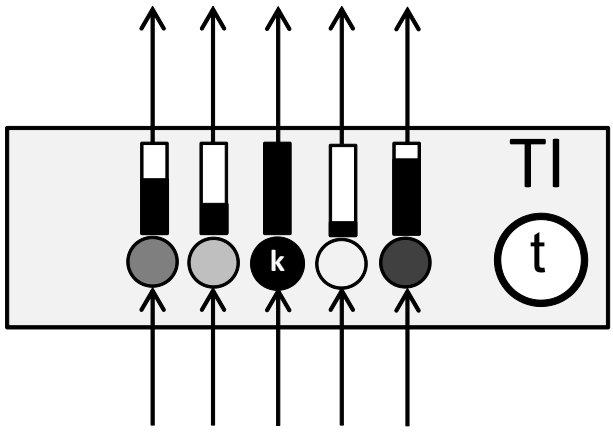

Figure 2.6: Temporal Integrator subnetwork. Note that the thermometer bars in the figure graphically represent an example of recency gradient encoded by the layer.

events. The S Fuzzy ART performs a classification of each pattern as it appears, each one resulting in the activation of an S-$F2$ node to which the network assigns to the pattern. (Note that references to Fuzzy ART internal layers will follow the notation of X-Y where X is the network's label and Y is a structure or layer in the module.) The S-$F2$ output of the S Fuzzy ART are directed as input into the TI layer, over time creating a recency gradient across its nodes. Following the presentation of the input pattern to S, the current real-valued recency gradient is directed to the input of the upper L Fuzzy ART that performs its classification. The classification processing of the S and L Fuzzy ARTS results in a collection of learned template patterns based on their respective input sequences. Through this hierarchical process, the sparse (winner-take-all) L-$F2$ outputs encode temporal sequences of S input patterns. The two ART units' vigilance parameters S-ρ and L-ρ control the granularity of their respective codes. As an option, the TI can be reset following an L Fuzzy ART recruitment of a new L-$F2$ node.

This network may recall temporal sequences through a top-down process illustrated in Figure 2.7 as reciprocal arrows. The top-down recall process is described in greater detail later in this document in section 2.8.4. During recall mode, an L-$F2$ node is activated by higher-level circuitry, reading out its template T_{Lh} into TI.

Using circuitry not shown in this figure, the TI activates the S-$F2$ layer in the same temporal order as it was produced, reading out a sequence of templates T_{Sk} into the S-$F1$ layer [74]. This "movie" of templates constitutes a short-term episodic recall.

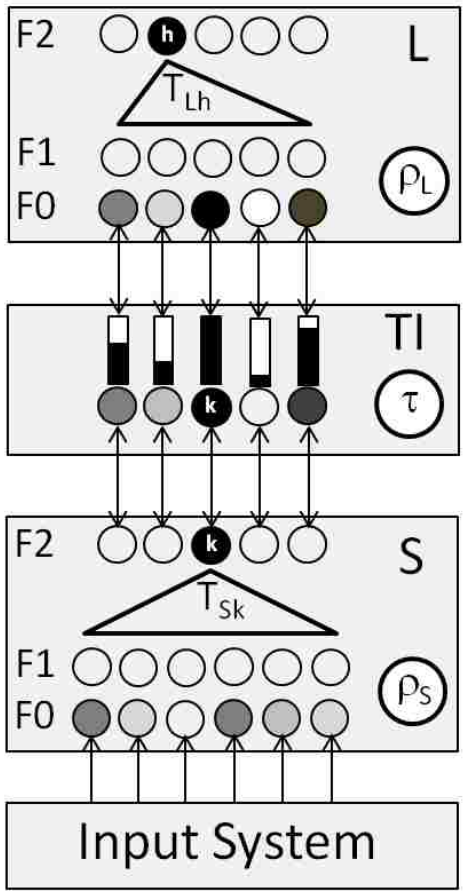

Figure 2.7: Temporal Sequence Learning Subnetwork. The lower Fuzzy ART is labeled S (short time scale) while the upper one is labeled L (long time scale). Layers with in these are referenced in the text as, for example, L-$F2$ or S-$F0$.

2.8.3 Background of Artificial Neural Temporal Methods

Vila [75] makes the case that time is ubiquitous in any activity requiring intelligence. While there are useful problems that can be solved without considering time, see any

number of static recognition or synthesis tasks, Vila gives several good examples of time dependent tasks: medical diagnosis [76] and explanation, planning, industrial process supervision, and natural language understanding [77] to name a few. Additional tasks that require the encoding of time include speech recognition [78] and synthesis, and time series prediction [79]. Some architectures [80] even transform a static problem into a time varying problem. Overall, the inclusion of a temporal aspect yields a sequence learning problem of which there are four fundamental types: sequence prediction, sequence generation, sequence recognition, and sequential decision making [81].

A research question then is, how can time be incorporated into the functionality of a neural network? One approach is to explicitly encode the entire desired temporal sequence as a single input. NETtalk [82] attempted to use such a spatial representation to convert words into their representative phenomes. However, such a spatial approach is problematic in several ways. First, it is not biologically plausible as brains do not buffer an entire sequence before processing the signals in any way. And secondly it requires a large enough input field to encode the entire sequence [83].

Alternatively, time may be incorporated in an implicit manner such that it is interleaved within the processing of the network. The addition of a memory component transforms the output of a neural network into a function of time [84]. While the synaptic weight values of a neural net inherently give the network a form of long term memory, they have a shortcoming in that they can only encode a single state of information about a gives set of neurons as opposed to a sequence of activations across time. Short-term memory modules allow neural activation values to be amassed across common episodic timescales. The buffering of prior node activation values produces a temporal representation ability within the network.

One possible approach for encoding activation sequences through time to enable temporal processing of data are time-delay neural networks (TDNN). Waibel, Hinton,

and Lang first described TDNN while attempting to classify sound spectographs [85]. A TDNN is a feedfoward network which utilizes time-delay units to replicate hidden layer and output layer neurons across time. A related approach is the tapped delay line [86], which can be considered as a TDNN with the time-delays all found on the input layer. The TDNN concept is best suited for classification of symmetric temporal patterns of fixed feature dimension [84].

A more generalized approach utilizing feedfoward networks and time-delays is the time lagged feedfoward network (TLFN). There are two categories of TLFNs, focused and distributed. The focused variant consists of an array of temporal delay filters serving as inputs to the desired neural network. The universal myopic mapping theorem developed by Sandberg and Xu mathematically proves the existence of a TLFN which can approximate any shift-invariant myopic dynamic map [87]. Furthermore, this architecture can be applied to arbitrary neural networks with known learning rules. Alternatively, the distributed variant overcomes the time-invariant restriction of focused TLFNs, which is a constraint of the universal myopic mapping theorem. The distributed approach employs a finite-duration impulse response (FIR) filter as opposed to simple linear filters. Furthermore, as a tradeoff for alleviating the time-invariance requirement, the distributed approach requires more complex learning rules such as the temporal-back-propagation algorithm [84].

Another approach to neural network temporal processing is through recurrent networks which implement feedback loops as a means of temporal buffering [88] [89]. Jordan networks, [90] also known as sequential nets, are one such approach by which the output layer loops back to state unit modules which buffer the feedback and pass these values into the hidden layer. Similarly, an Elman network offers feedback which is buffered through state units; however the recurrence occurs between the hidden layer and the state unit layer directly. For both architectures, the back-propagation algorithm works by treating the state unit layer as extra inputs to the hidden layer. An extension to the back-propagation algorithm for recurrent neural

networks is known as back-propagation through time (BPTT). The BPTT approach is to unfold the recurrent network by duplicating the recurrent components as many times as necessary so that an equivalent feedforward network exists and the standard back-propagation algorithm may be applied [91].

A critique which may be applied to the aforementioned temporal neural network models is that they are primarily supervised off-line learners. Off-line learning models require that any time a new sequence is to be learned the entire system must be retrained. Furthermore, the laborious training time required to reach stability is compounded by the temporal aspect in the sense that training data cannot be randomly ordered every epoch to quicken the process, or if so there would be no temporal meaning associated with the training sequences. Alternatively, Bradski et al. [92] have developed a class of dynamically defined working memory neural network models called sustained temporal order recurrent (STORE) models. These models encode the temporal order of arbitrary sequences of items. The STORE class of models has progressed incorporating additional features in the later variants. The simplest one-level variation, STORE 0, is "adequate for working memory coding and recall, provided that input durations are restricted" [92]. Expanding upon this capability, the two-level model STORE 1, "encodes temporal order for input sequences whose items are not repeated" [92]. Various modifications may be applied to alter the performance of STORE 1. One particular modification is the incorporation of a decay term in STORE 2 which "adds a parametric degree of freedom to the control of relative sizes of working memory representation" [92]. And finally, STORE 3, "extends system capabilities by allowing both repeated and non-repeated item sequences to be encoded and recalled" [92]. Incorporating the STORE working memory model into a larger model, Bradski et al. created an ARTSTORE model by cascading an invariant processor, an ART2 module, a STORE model, an ART2 module, and an Outstar network [93]. Although the STORE class of models has similar properties to our architecture, in general it is computationally more complicated. As will be

described in more detail in the next section, our approach requires a single level whereas the STORE approach typically requires three. This simplification can be significant when incorporating the temporal block into a large complex architecture which may employ hundreds of temporal modules. As follows, we present our unsupervised online learning architecture for temporal semantic encoding in combination with ART modules.

An additional example of an on-line temporal learner can be found in the work of Willams and Zipser [94]. In that example though, non-local connections and significant computation are required. In the following, an unsupervised online method for temporal semantic encoding is encoded. Our method introduces a computationally slight addition to the non-temporal system, and requires only additional local connections. The constraint to local connections allows this method to modularly model functional subsections as found in biological cortex [95].

In addition to the above considerations, this method is notable in that it can plausibly encode temporal sequences of arbitrary scale. Methods that rely on delay lines or recurrent connections would need arbitrarily long delays or arbitrarily many recurrent connections to encode time at arbitrary scales.

The following details a method to perform a structure preserving map of observed temporal episodes into artificial neural tissue. Sensory input is received as a continuous temporal series, but memory is encoded as a parallel spatial distribution of synaptic weights. This investigation develops the concept of adding temporal integration between layers of ART units. We begin with each architecture layer composed of a single ART module. An array of temporal integration units follows the ART module. If we consider an input vector as a vector of temporal data streams, i.e. input values varying through time, then the output of the temporal integrator module is a vector of values that reflect the recency of input values at each position. A temporal series of sensory signals that forms input to the first ART layer is encoded into a temporal series of output vectors. By virtue of winner-take-all competition at

the output layer of ART, these output vectors have only one active element. A series of distinct ART output vectors is temporally integrated into a pattern of activations that, in conjunction with the stored ART templates, represents the original series of sensory inputs. This pattern, output by the temporal integration array, can be stored in an ART module at the next layer. By using multiple ART modules at each level and multiple levels, this scheme can form arbitrarily detailed and distributed representations of episodic inputs.

The method of the experimentation upon a temporal episode architecture component is to encode temporal semantic data as a recency gradient of classifications. ART is used as the classifier that creates an active output on a certain F2 node for any given input vector. Each element in the output vector of ART F2 nodes is connected to the input of a leaky temporal integrator. It is important that the integrator be leaky, otherwise the output would continually increase over time (assuming continual input greater than zero), eventually saturating. A general leaky integrator is modeled as Equation 2.12.

$$\frac{dy}{dx} = -ay + x \tag{2.12}$$

$$y(n+1) = (1-a)y(n) + x(n) \tag{2.13}$$

A discretized (using Euler's method with an arbitrary sample period of one time unit) version of the leaky integrator is implemented in Equation 2.13. $1 - a$ is the decay constant. In both equations, x is the input and y is the output.

A comprehensive example of this method's operation is provided next. The following example assumes three arbitrary sensory input vectors. For the example, let these sensory inputs create a grounded, stand-alone, example. However, if the sequence of inputs over time were internal cortical activation patterns, the example

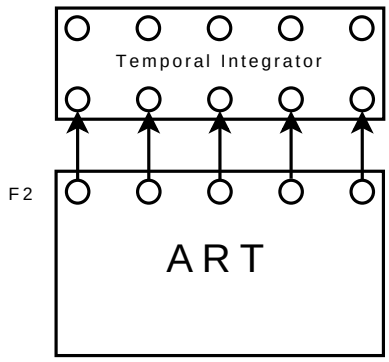

Figure 2.8: ART with temporal integrator. Each $F2$ node has a temporal integration element attached.

could be describing an additional functional level of cortex, i.e. internal cortical activations instead of external sensory activations.

Let there be three distinct sensory input activation vectors A, B, C that form a temporal semantic sequence which will be encoded. Inputs feed into an ART module (see Figure 2.8). Each input results in a different active node output on the ART. By placing a leaky temporal integrator on each ART output node, a temporal sequence of inputs is encoded as a single real valued vector. The integrated vector is a recency gradient, where the order of element amplitudes (from low to high) represents the order of occurrence of the input vectors (from oldest to most recent). As mentioned, the value of a given integrator output node will decrease over time. As a result, the farther in the past a given input was observed, the smaller a value the corresponding integrator output will have (until at some small activation level, the integrator output is lost in the noise of the system).

If the first input pattern is 0110, let a corresponding output of the ART be [1 0 0], considering only the first three output nodes for simplicity. The temporal integration array initializes to [0 0 0] and upon receiving the first ART module output, becomes [1 0 0]. At the second timestep, the ART module gets input B and outputs [0 1 0]. The temporal integration array decays down to [0.707 0 0] then receives the timestep two ART module output to result in the temporal integration array output

Temporal Integrator Values	ART Input	ART Output	TInt Output
Time 0		0 0 0	0 0 0
Time 1	A	1 0 0	1.0 0 0
Decay 1-¿2		0 0 0	0.707 0 0
Time 2	B	0 1 0	0.707 1.0 0
Decay 2-¿3		0 0 0	0.5 0.707 0
Time 3	C	0 0 1	0.5 0.707 1.0

Table 2.1: Temporal integrator module operation illustration

of [0.707 1.0 0]. At the third timestep, the ART module receives input C and gives output [0 0 1]. The temporal integration array decays down to [0.5 0.707 0] then, with the new input of the ART output, becomes [0.5 0.707 1.0]. This process is captured in Table 2.1.

Finally then, the temporal input sequence A, B, C is encoded at a single point in time as the vector [0.5 0.707 1.0]. To the level of detail that the ART categories have generalized afferent input vectors, [0.5 0.707 1.0] encodes both the temporal and semantic information of the input sequence. The temporal information is the order of the input sequence. The semantic information is, in this case, sensory activations that we labeled A, B, and C. Note that the semantic information is only preserved through the connectivity of the temporal integration array to the ART module and through the templates in the ART module.

Extending the architecture, the vector [0.5 0.707 1.0] can now be encoded by a level 2 ART, see Figure 2.9, to uniquely represent the sequence A, B, C. Similarly, B, A, C would result in a level 2 ART representation of [0.707 0.5 1.0], given that the outputs of level 1 ART were [0 1 0], [1 0 0], [0 0 1] in sequence.

Given that the level 2 ART representations are unique (to the level of precision determined by the ART operational parameters), a top-down recall operation can recover the original sequence A, B, C from the level 2 ART encoding of [0.5 0.707 1.0]. For example, the temporal integrator output of [0.5 0.707 1.0] may result in the activation of the first F2 output on level 2 ART. Then, given an augmented ART with

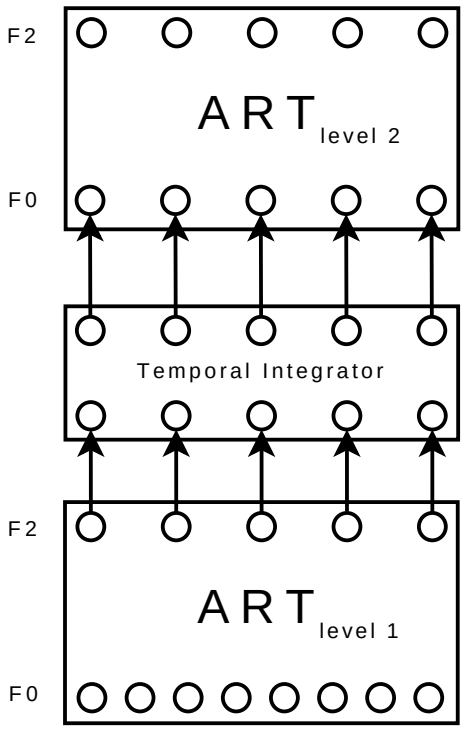

Figure 2.9: ART with temporal integrator followed by ART. This architecture element creates classification of temporal sequences of classifications of neural activation patterns.

top-down recall ability, as well as a playback mechanism for the temporal integrator, subsequent top level stimulation of the the first F2 node on the level 2 ART will result in the sequence A, B, C being played back at the system input level.

To enable the use of this temporal integration mechanism, ART is supplemented by specifying a top-down behavior. When an $F2$ node is stimulated from above, it plays down the associated template's activation levels to the ART input layer. Template activation level adjustment is where all memory storage in ART occurs, so playback of a given template represents recall of one memory component at the scale of that ART unit.

To enable recall, the bottom-up temporal integration scheme is supplemented with a top-down behavior. As the output of a temporal integrator array is a pattern encoding the order of input activation, recall of that gradient should play back the

temporal integrator inputs in the observed order. When top down signal propagation places a pattern on the temporal integrator array from above, the array will first activate the input corresponding to the lowest value in the pattern, then the next lowest, and so on. This behavior will play back the input activations in their original order. As a manner of implementation, it can be imagined the top-down stimulation of the temporal integrator array sets a threshold for each element of the array. The integrators then start integrating up from zero and fire the associated input node when their internal value reaches the threshold. The lowest threshold will be reached first, which is correct because it represents the input that occurred farthest in the past, the input which has decayed farthest, and therefore the input that should be played back first.

2.8.4 Top-Down Recall

Recall played a central part in the operation of embodied neural agent, so before embodied experimentation it was confirmed that the recall mechanism functioned as desired. Figure 2.10 shows the sequence of simulated foveal input data. The first level ART forms templates to represent the input visual data symbols. The temporal integrator forms spatially encoded patterns to represent sequences of symbols observed by the first layer ART as passed on by the first layer ART's $F2$ node activation. The second layer ART forms templates that represent the patterns output by the first layer temporal integrator array that represent temporal sequences of input symbols. In this case, the timescales are such that the second layer ART captures temporal integrator patterns corresponding to three input timesteps. Figure 2.11 shows the recall of one such sequence.

Note that the recalled sequence is not an exact copy of the input sequence. The plus and diamond symbols have been aliased to a common symbol. This aliasing occurs because in the input to template matching used by ART, the plus and diamond symbols are sufficiently close for classification to the same $F2$ template. An easy

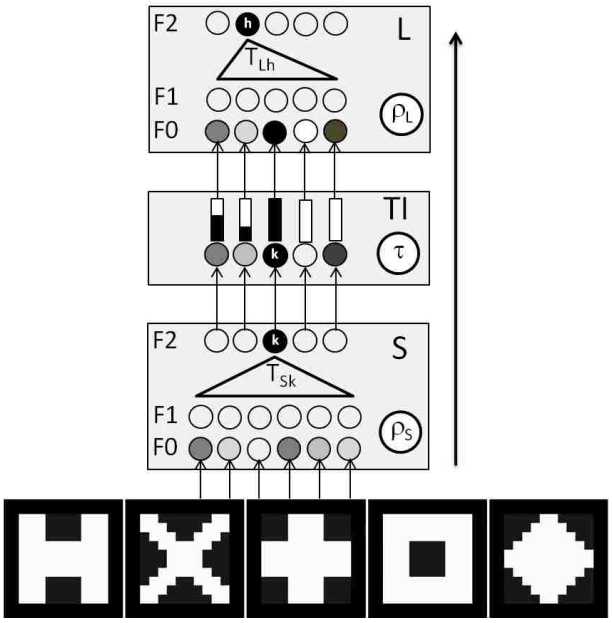

Figure 2.10: Memory and recall input sequence.

parallel can be drawn in human memory formation. To the non-expert, transient observation of either a grey Toyota Corolla or a grey Honda Accord is likely to form an aliased memory of "grey import sedan." In ART, the vigilance parameter determines how close two inputs must be to be classified into the same template. The vigilance parameter was set arbitrarily at 0.8, in a range of 0 to 1, for the experiment. In our architectures, the vigilance parameter of the ART units is an independent variable that can be used to tune the performance of the network. One of the trade offs inherent in tuning with the vigilance parameter is memory space versus precision, greater vigilance will form a greater number of more precise templates. Memory space in our model is analogous to cortical tissue. We will be able to tune a part, or all, of our model for more precise memory formation, but only at the expense of using up more of a limited resource.

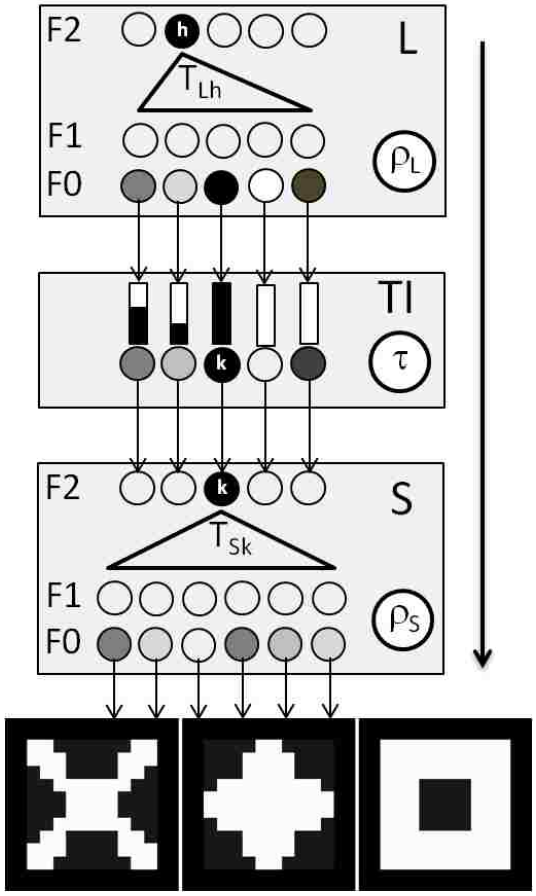

Figure 2.11: A sample memory and recall output sequence.

Background has now been provided for the methods that will be implemented to test the theories put forth as embodied artificial neural architectures. The next chapter, Apparatus, describes the neural architecture which is built and placed into a simulated embodiment.

Chapter 3

Apparatus: Embodied Environment and Neural Architecture

This chapter describes the virtual apparatus which enabled the experimentation described in following chapters. The chapter begins with a description of the neural simulator which is used to run experiments, followed by a description of the real morris water maze apparatus upon which the virtual water maze is based. Then begins the descriptions of virtual components. The simulated rat agents's body is described, followed by the computational neural architecture which comprises the simulated brain. Lastly described is the simulated environment in which the rat agent computationally embodied.

3.1 Neural Simulator

eLoom [96] is an open source software tool that allows users, through the use of a specification language, to construct and execute a C language program that simulates arbitrarily connected, hierarchically organized neural networks and associated

simulation behaviors. The specification language supports five component types: modules, nodes, node arrays, directed connections, and directed connection arrays. Nodes and connections may be grouped into arrays or modules, and modules may contain other modules. All components have user-specified simulation functions and data storage.

The eLoom parser takes as input the specification file and constructs a system composed of efficient C language data structures for the designated network components, each containing function callbacks. The function schedulers may be application dependent and therefore may require customization by the user. Upon completion of a simulation, eLoom has the capability to regenerate the network's specification file (useful for self-modifying neural networks) and to save an engram of all network components' states. This allows a simulation to be continued from the point it stops. Each Flatworld agent has an independent eLoom-specified neural system that can access all agent sensors and actuate all agent motor functions. The code for each agent runs in a separate thread, allowing for efficient execution on modern multicore computer architectures.

3.2 Morris Water Maze

The Morris water maze is one of the most widely used tasks in behavioral neuroscience for studying the psychological processes and neural mechanisms of spatial learning and memory [97]. The physical Morris water maze task consists of placing rats into a large circular pool of water from which they can escape onto a hidden platform. The platform is hidden by virtue of being just under the surface of the water, where the water is opaque due to addition of powdered milk or some other substance. As such, the platform offers no visual cues to guide escape behavior. While rats could theoretically escape by swimming randomly or in unsystematic search paths within the pool, in practice normal rats learn to swim directly toward the platform from an

arbitrary starting position at the edge of the pool [98].

The following section describes the neural architecture which incorporates the embodied hippocampal model. This system is the one which will be implemented in a simulated embodiment. The embodied hippocampus incorporates the neurophysically based design scheme from the stand-alone hippocampus with architectural changes to support sensory-motor episodic memory moderated by the dopaminergic system.

3.3 Embodiment

This section describes the computationally embodied neural architecture. The computational model and simulated environment was produced using the network specification and simulation package eLoom [96]. In the first subsection, the treatment of motor output for the embodied agent is described. Then the link between behavior and perception is discussed, followed by the link between motor output and sensory input. The dopaminergic subsystem, which modulates episodic memory binding based on reward and expected reward observation, is covered next. Finally, the total architecture and its operation is described.

3.3.1 Embodied Agent and Flatworld

Flatworld [96] is a general-purpose two-dimensional agent-based modeling tool for the training and testing of one or more agents. The 2D world is populated with geometrical objects, with which an agent may interact. The agents and objects have physical properties such as shape, color, food value or potential energy, position and velocity, and mass. An agent has a round body with visual and contact sensors as illustrated in Figure 3.1.

In the simulations, agents have sensors, both internal and external, and actuators. Sensors include artificial eyes, skin contact sensors, and muscle actuator states.

Actuators include artificial muscles for movement within the world (forward- backwards, left-right, change in heading). Light signal strength diminishes with distance, similar to the real world. Finally, each agent has a controller that reads sensor data, both internal and external, and produces control outputs to its actuators. The control of agents may be individually customized through the use of "personality" parameters. In this study, the controller is the neural architecture.

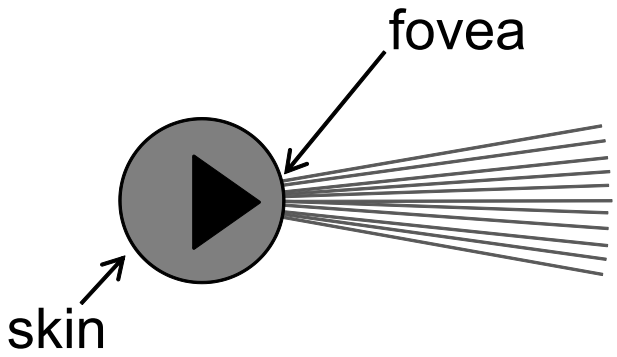

Figure 3.1: A representation of an agent in Flatworld.

3.3.2 Neural Architecture

The complete neural architecture is described in this section. It is composed of a small number of subnetwork components including Fuzzy ART [64], Fuzzy LAPART [69] [99], Temporal Integrators [73], gating logic, and sensors/actuators commensurate with those of the agent described above. Fuzzy ART, Fuzzy LAPART, temporal episode encoding methods, and the dopaminergic system were introduced in the Background chapter. The following describes how these subnetworks interconnect and interact to form the distal reward neural architecture.

Figure 3.2 illustrates the overall distal reward architecture that is composed of the previously described subnetworks. The agent has a visual sensor composed of an array of color brightness detectors, proprioceptive sensors that measure motor activity, touch sensors that detect contact with objects in the world, and finally

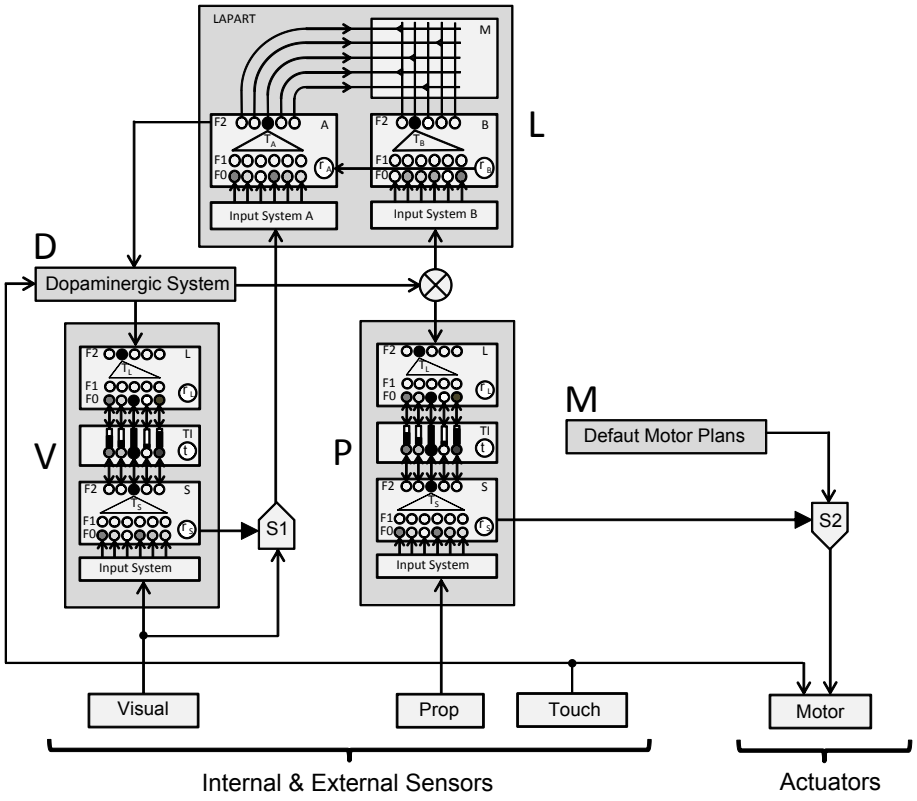

Figure 3.2: The complete neural architecture. The boxes along the bottom represent sensors and actuators available to the agent. The Visual sequence learning track (V) is on the left while the Proprioceptive sequence learning track (P) is in the middle. The LAPART (L) system performs associative memory functions. A simple dopaminergic system (D) provides reward dependent control of the architecture. Finally, the Default Motor Plan (M) component generates random-walk motor plans for exploratory behaviors. The pentagons labeled $S1$ and $S2$ are circuits that switch one or the other input bundle through to their outputs as follows: If the input indicated by the solid arrow is active, then it is directed to the output, otherwise, the open arrow input is directed to the output. The circle with a cross is a gating circuit that controls the bidirectional flow of signals.

motor actuators that direct the speed and direction of the agent in the world. The neural architecture is divided into five major components: 1) V, the visual temporal sequence component, 2) P, the proprioceptive temporal sequence component, 3) L, the Fuzzy LAPART association component, 4) D, the dopaminergic component, and 5) M, the default motor plans component. The latter component generates random

walk motor commands in the absence of a learned motor plan recall. These components are interconnected with connection bundles, some of which are modulated by switch or gate circuits. A switch circuit directs one or the other input bundle to its output, while a gate may block the bundle with a single control connection.

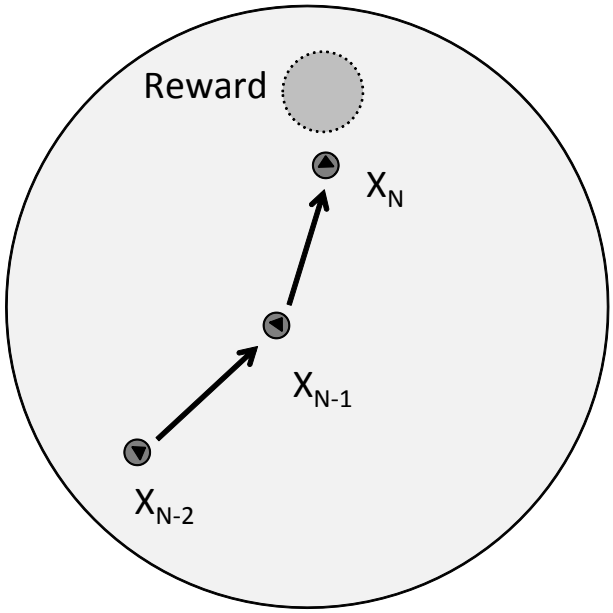

Figure 3.3: An example of the agent moving between way-points to arrive at the reward platform. Either a learned or a default motor plan moves the agent from one way-point to another.

How does this architecture learn to find its target through distal reward learning processes? To answer this, we will use the Morris Water Maze as an example scenario. In the first learning trial, Trial 1, the agent begins its existence at a release point in the tank and, having learned nothing at this point, it executes a default motor plan that causes exploratory behavior. See the lower right-hand portion of Figure 3.2. Referring to Figure 3.3, in Trial 1, the release and reward locations are X_{N-1} and X_N respectively. During the movement between these two points, the V and P components are actively capturing perceptual sequences encoded as $VL\text{-}F2$ and $PL\text{-}F2$ node activations. (Note that references to Fuzzy ART internal layers within

major components will follow the notation of CX-Fn where C is the component label, X is the label of a Fuzzy ART internal to that component, and Fn is one of its layers.) When the agent ultimately encounters the hidden platform at location X_N, the touch sensor signals the D component of the presence of a reward in the environment. The D component is designed to activate two possible processes: 1) retrospective learning [100], and 2) prospective action. Retrospective learning is the process of learning from the past, while prospective action is the process or planning for the future. In this case, the agent has encountered the reward for the first time and therefore the D component enables only the retrospective learning process.

The Trial 1 retrospective learning process is graphically illustrated in Figure 3.4. This form of learning occurs in the L (LAPART) component of the architecture with support from the D component. Due to D component activation, the V component undergoes episodic recall using the VL-$F2$ node active at the time of the reward. This recall would, if allowed to completely unfold, play back a temporal sequence of visual templates in the VS-$F1$ layer, learned from the sequence of visual perceptions experienced during the motion from X_{N-1} to X_N. For the purposes of learning a distal reward, only the first (oldest) visual template is recalled and forwarded to the A-side of the L component through switch $S1$ (right center of Figure 3.2). In parallel with this, the D component gates the PL-$F2$ node active at the time of the reward into the B-side of the L component. This node codes the proprioceptive perceptions experienced during the motion from X_{N-1} to X_N. The A and B-side Fuzzy ARTs in the L component perform their self-organizing classifications, and if a lateral reset does not occur, a lateral priming connection is learned between LA-$F2$ and LB-$F2$. That is, a strong synaptic weight is learned in the L-M association matrix creating a priming between the winning LA-$F2$ and LB-$F2$ nodes. This creates an association between the A template representing the visual scene at location X_{N-1} and the proprioceptive plan representing the motion from there to the reward at location X_N.

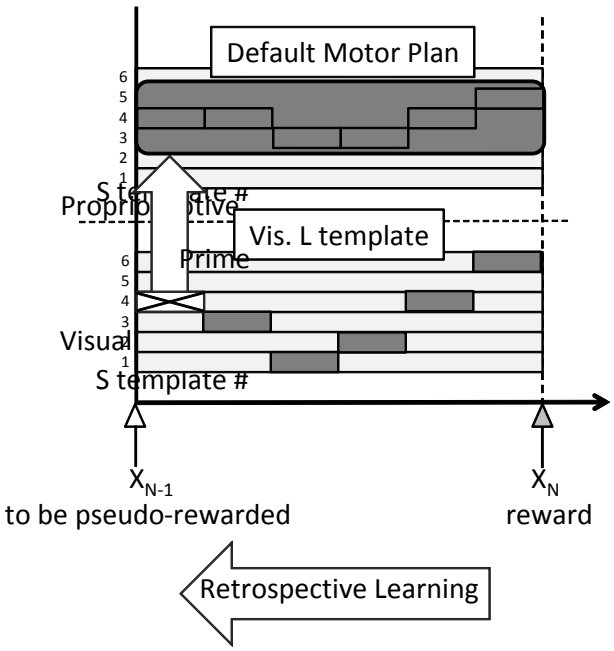

Figure 3.4: Illustration of retrospective learning, when a reward is detected at position X_N. A priming connection is learned between the oldest VS-$F2$ template captured near position X_{N-1} and the PL-$F2$ template that encodes the current proprioceptive sensory sequence generated by default motor plan exploration. This priming occurs within the L component. In the future, the location that resonates with this VS-$F2$ template will provide a "pseudo" reward for all previous proprioceptive sensory sequences leading to that location. See Figure 3.3.

For the second trial, Trial 2, we assume that the agent is released at location X_{N-2}, illustrated in Figure 3.3, and having learned nothing about this new location; it again executes a default motor plan that causes exploratory behavior. In this example, we assume the default motion moves the agent to location X_{N-1}. Note that when the D component is not active, raw visual patterns are simultaneously directed to the A-side of the L component through the $S1$ switch as well as to the V component. If at some point during motion a visual input pattern resonates with an existing LA-$F2$ template, then two actions will occur, illustrated in Figure 3.5.

First, the D component is activated through the connection from the LA-$F2$ layer, signaling that the agent is at a familiar visual location previously associated with a rewarding experience. We refer to this as detecting a "pseudo" reward in

the sense that the agent is aroused as if an actual reward has occurred because it now "knows" how to get to the reward from this location. This leads to a new retrospective learning process as describe in Trial 1, creating an association between the a template representing the visual scene at location X_{N-2} and the proprioceptive plan representing the motion from there to location X_{N-1}. See the left half of Figure 3.5 for an illustration of this process. Upon completion of retrospective learning, the D component becomes inactive.

Second, through an association learned in Trial 1, the active LA-$F2$ node resonant with the current visual input pattern at X_{N-1} will laterally prime the B-side of the L component. This will cause the B-side Fuzzy ART in LAPART to the read out the associated PL-$F2$ node that encoded the proprioceptive sequence for motion from X_{N-1} and X_N. This information is directed through the gate to into the PL-$F2$ layer of the P component, initiating an episodic recall of the proprioceptive sequence. See the right half of Figure 3.5 for an illustration of this process. During this recall, PS-$F2$ templates are activated in the learned sequence priming the PS-$F1$ layer with time varying proprioceptive signals. Finally, these signals are directed to the motor actuators through switch $S2$, blocking the default motor plans.

A unique aspect of this approach is the use of proprioceptive recall to replay motor function, reminiscent of Common Coding Theory [101] [102]. As the recall occurs, PS-$F2$ templates are channeled to motor actuators in the same sequence as they were learned, thus reproducing the motion from location X_{N-1} to X_N. This assumes that for every actuator there exists a proprioceptive sensor and that sensors are appropriately scale to the stimulus. At the end of the recall, the agent will find itself at the reward. By recursive application of Trial 2 notions with multiple agent release locations and orientations, the architecture will begin to tessellate the water tank space with pseudo-reward locations, creating chains of proprioceptive sequences to move to the reward location.

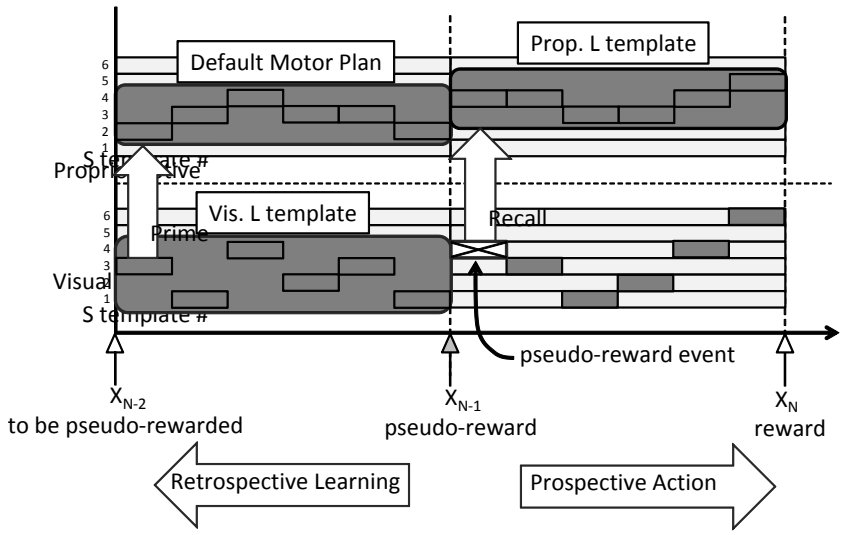

Figure 3.5: Illustration of both retrospective learning and prospective action that occurs when the agent arrives at a distal "pseudo" reward position X_{N-1}. At this point, the agent learns the association of the oldest V S-template with the current PL-template proprioceptive sensory sequence.

3.3.3 Architecture Parameters

Several selectable parameters influence the computational architecture behavior. These parameters include:

- ART module vigilance parameter.
- Temporal semantic representation time constant of integration.
- Angle of visual sensory arc.

The computational neural architecture contains numerous ART modules, each module with a potentially unique vigilance value. A large amount of study could be devoted to characterizing embodied behavior under different combinations of vigilance values. In the greater context, only the vigilance values which results in approximately optimal behavior are interesting. Recall that vigilance values control the generalization level of output classes. It can be assumed that biological neural ensembles performing classification function, such as simple cortical cells, have

approximately optimized their level of generalization through some form of feedback. This occurs outside the context of experiments performed with biological rats. Likewise, while the optimization of generalization level has occurred for the computational architectures, it is outside the context of behavioral experiments and is not empirically reported on.

Temporal semantic representations in the computational model are formed as temporally integrated recency gradients of classifications of neural activations. These neural activations are perceptually grounded in sensory semantic information. The time constant of discrete integration is a model parameter which controls how quickly each component instant in a temporal memory decays away. Too quick of a decay would remove the temporal aspect of the temporal semantic representation because all neural activation history would decay below noise at the first step. Too slow of a decay would result in poor chaining of expected reward cued motor paths. It is the chaining of expected reward observations which allows a computational rat to form navigation skills that generalize across the physical space of the tank. With only a single long motor path, the rat might perform very well at reaching the platform from a single release point. However, one of the behavioral task is a transfer learning generalization task, when training occurs from one release point and testing from another. Performance would be expected to be very poor in the transfer learning task if temporal decay were too slow.

Through simulated embodiment, visual cortex is activated due to environmental cues. Afferent visual stimuli are centered in the direction that the computational rat is facing. Biological rats have a vision field of roughly 210 degrees. Detailed modeling of biological visual input would be fairly complex. While moving in the morris water maze tank, rats constantly rotate their heads back and forth. As a result, it can be assumed that their sensory cortex integrates visual input from nearly 360 degrees around the rat. Not all of this environment will be sensed to the same resolution though, the foveal region at the center of the retinal sensor field has a higher density.

The general consequence of this higher foveal density is that whatever the eye is looking at will be seen in more detail. Based on this information, an implementation decision was made that environmental information would be present in the sensory cortex in a high resolution form for stimuli straight ahead of the rat and in a low resolution form from stimuli all around the rat. The angle which is considered straight ahead is configurable. In the embodied model, it is fifteen degrees on each side of center, or thirty degrees of total arc .

3.3.4 Expected Reward Visualization

Figure 3.6 is a visualization of sensory observations bound to expected reward. Colored arrows mark location and orientation of sensory observations that have been chained back, through experience, to reward. To form the visualization, a rat model was embodied in simulation to build up sensory memories and bind them to expected reward. These reward bindings were then probed for a match to the sensory stimuli that a rat would observe from each point in a grid superimposed on the simulated environment. At each point, observations were considered in two degree increments of orientation. An arrow is drawn at points and orientations where a match is found with the rat model's bound memory. The color of the arrow indicates expected reward level. Red arrows indicate higher expected reward, and proceeding down through the spectrum, blue arrows indicate low expected reward. Figure 3.7 provides a close-up view of the visualization.

As one might expect, given the above description of expected reward binding, bound locations that are close to the platform have high expected reward. These near platform locations are likely one motor path segment away from the platform and actual reward.

In Figure 3.6, we see red arrows in the bottom left portion of the tank, far from the platform. We hypothesize that these formed as a sensory alias of the reasonable red arrows near the platform. The term sensory alias is used here in the

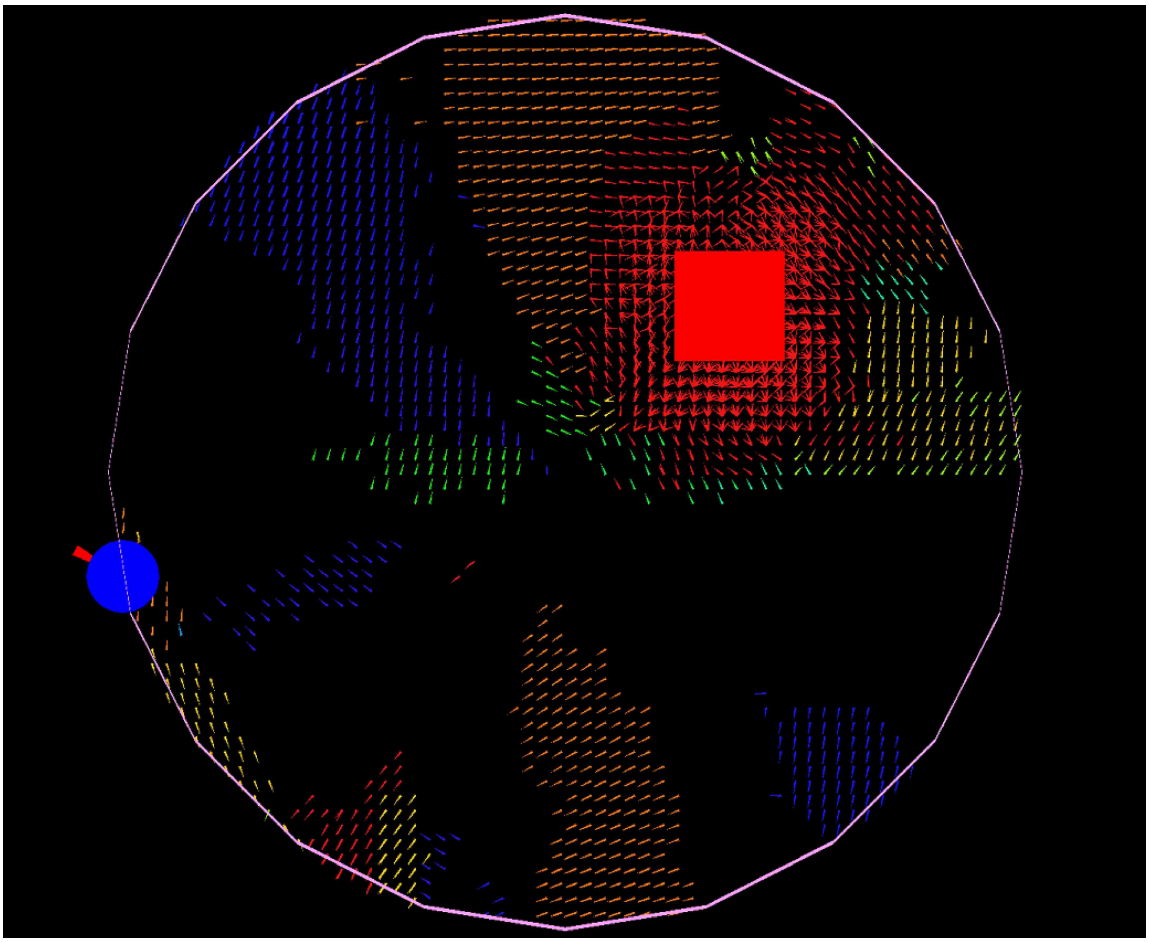

Figure 3.6: This figure visualizes the expected reward content of episodic memories bound in the embodied computational hippocampus. Arrows represent bound episodic memories with a corresponding expected reward level. Red arrows indicate high expected reward and blue arrows low. New episodic memories are bound when observations linked to expected reward occur. During exploration if the computational rat were to become located at the tail of an existing arrow, new arrow formation would occur.

sense of a discrete sample sensor which causes aliasing between frequencies which are greater than half the sampling frequency. When we probe the model's memory for matches to sensory stimuli with bound expected reward, matches can occur for locations that were never actually bound to memory. The lower left portion of the tank simply appears to the rat model as similar to some portion that is near the

Figure 3.7: Zoomed expected reward visualization.

platform. Behaviorally, this might lead to apparent confusion if the rat model tries to follow a false memory.

In Figure 3.7, we see some red arrows that are oriented tangential to the platform. These arrows appear to indicate that non-optimal paths might be taken. The model does not purport to solve the distal reward problem optimally though. It is generally reasonable to suggest that if the rat is very near the platform, movement in any number of orientation can still lead to reward in fairly short order. Also, consider that as the rat can not see the platform, there is very little difference between the sensory stimuli from being just north of the platform, where a south-west arrow points directly to reward, and the sensory stimuli from being just around the corner, to the west, of the platform where a south-west arrow now appears to be in error. The general trend of the visualized arrows is the most interesting result.

The expected reward visualization provides good evidence that the model works as intended. Behavior experimentation in the next section will provide further evidence that the embodied neural architecture behavior corresponds with the behavior

of embodied biological neural architectures, i.e. real rats.

3.3.5 Model Assumptions

Several assumptions have been made in the design of the computational neural architecture. Each assumption is explained in more detail in a paragraph after the list. The assumptions include:

- Resolution of the simulated environment is sufficient for experimental tasks.
- A two dimensional environment is sufficient to demonstrate distal reward learning.

An example of resolution of the simulated environment is the number of data points used to represent an object. A fully realistic simulation would need at least as many data points for a simulated object as there would be atoms in a real object. On the sensing side, a fully realistic simulation would use as many visual sensor elements as there are receptive cells in a corresponding retina. Computational load generated by the neural model constrains resolution. If unbounded computation were available, then sensor density equivalent to biological retinas could be implemented. As computation is of course bounded, the implemented sensory density is lower than the biology. This is compensated for by a simpler simulated environment as compared to the real world environment experienced by biology. It is asserted that so long as the modeled sensor density is sufficient to resolve between different features present in a given environment, then the attached neural system is receiving sufficient input resolution to function. A lack of successful distal reward behavior might indicate that input resolution is insufficient. Successful distal reward behavior will indicate that all elements of the computational neural system are at least sufficient. This same line of reasoning applies to other computational considerations, such as the maximum number of classification types allowed in ART modules. In these other cases, successful behavior will again validate simulation resolution assumptions.

It is assumed that biological rats have visual cortical functions which are capable of distinguishing between proximal and distal features in the environment. In large part, the biological ability to distinguish between proximal and distal objects is based on sensory input from a three dimensional world. Because the experiment simulations occur in a two dimensional world, exact neural processing equivalents would not function. Based on the assumption the distal and proximal object are differentiated between in visual cortex, that distinction is made implicitly in the model and separate cortical information streams are incident upon the input regions of hippocampus. These separate information streams are implemented as multiple one-dimensional features vectors from the observed environment. If this assumption is wrong, then the implementation should have been a single one-dimensional feature vector. Neuroscience literature provides many examples of distinct feature types extracted throughout sensory cortex from the original raw sensory input [103] [104] [105]. As such there exists evidence to support seperation of feature elements in the embodied architecture, which is a necessary consequence of the two dimensional environment.

3.4 Water Maze Environment

Figure 3.8 shows a visualization of the simulated water maze environment. The four cardinal rat release points are illustrated.

The simulated environment exists in two dimensions. Environmental simulation is performed through assuming a diffuse light source and then ray tracing visual sensor input. Each visual sensor element returns an intensity value for the first object to intersect a ray orgination from the sensor element. Objects are opaque, and as the environment is two dimensional, there is no existence of a large relatively distal item visible above a smaller relatively proximal item. The first item that the ray encounters in a given direction is the only necessary input to the returned value.

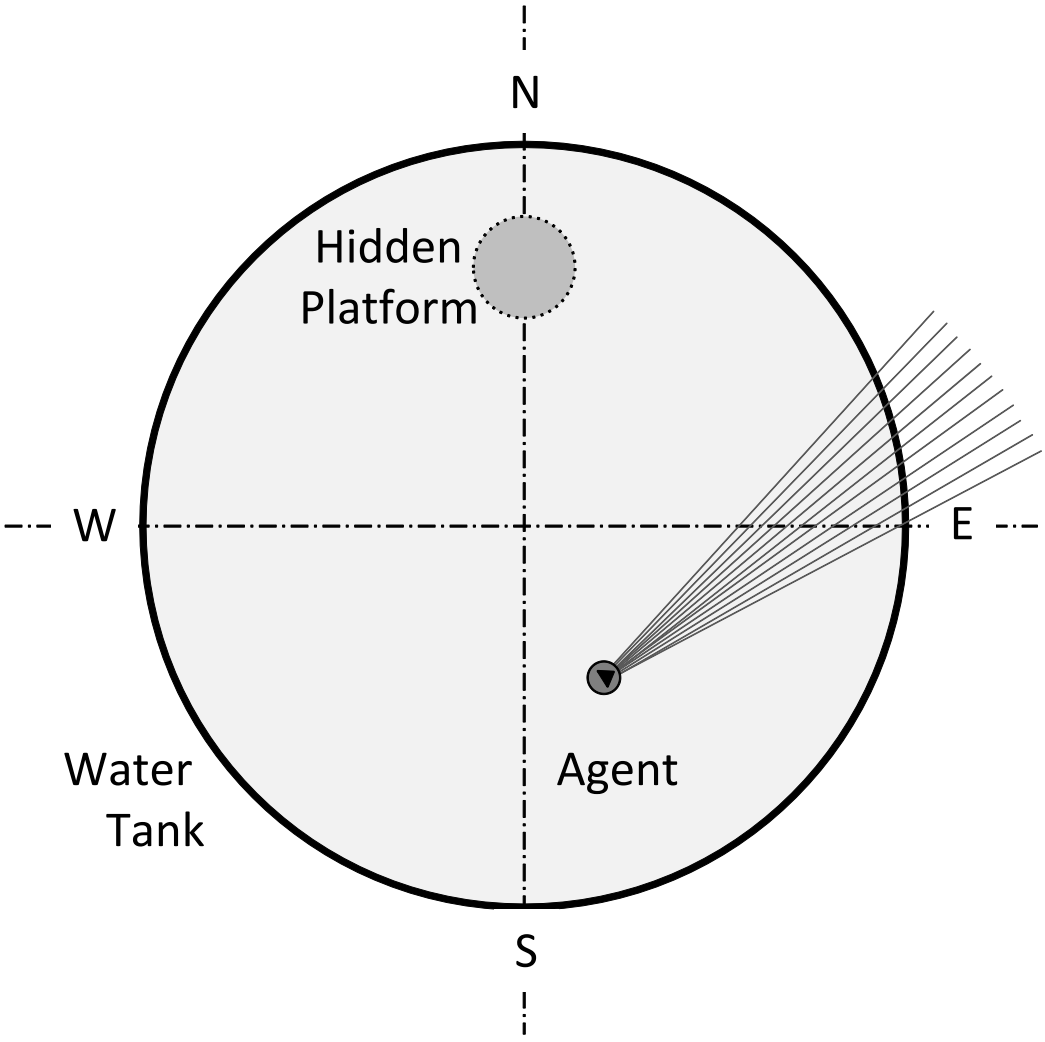

Figure 3.8: Illustration of water maze environment.

The sensed intensity value of an object is a function of the color of the object and the distance from sensor to object.

There are three objects in the Morris water maze environment in addition to the embodied agent; the tank, the room exterior, and the escape platform. The tank is a circle in which the embodied agent is constrained to remain. The tank is five hundred simulation distance units in diameter. The outer room is an iregular polyhedron that is roughly square in shape with many facets on each wall, these

facets make up the distal features of the experimental environment. The room is fourteen hundred simulation distance units across. Biological Morris water maze experiments are most often performed in a lab setting, where the distal features are bookshelves, sinks, and other items in the lab. The facets of the simulated room walls are functionally equivalent to the variety of visual cues which might be found in a given lab setting. The final object is the escape platform and is not visible to the agent. The escape platfrom is thirty simulation distance units across. The computational rat is twenty simulation size units in size.

Biological rats undergoing the Morris water maze task have numerous sensory cues available in addition to the visual and limited tactile cues that are simulated. Almost any lab setting will have olfactory gradients stemming from sources such as the rat cages. Auditory cues may exist such as forced air through a ventilation duct. Internal inertial navigation senses are also present. Together, these cues present the biological rat with a notably richer environment than the simulation provides. This makes the simulated Morris water maze task more difficult than the real task because there is less information present for differentiating locations in the simulation. For the purposes of experimental design it was important to err on the side of less sensory input as this places a greater burden on the learning system and results can not be explained away as due to extra sensory input beyond what biology would have.

3.5 Architectural Addition of Categorical Limits

Using category theory, Healy and Caudell have shown that a standard ART architecture (specifically ART1) can only partially represent the semantics of its input space [67]. Specifically its architecture does not contain the necessary node and connection structures that are required to unambiguously represent concepts. These important categorical structures are limits and colimits, and this section summarizes changes to ART which allow ART to represent categorical limits and colimits. More detail

is found in Appendix B. In the categorical limits augmented architecture, all ART units are replaced with the augmented version.

In category theory, a limit represents a unique property which all objects in a collection map from. The dual notion, a colimit represents a unique property which all objects in the collection map to. Notionally, the limit is a concept which a set of concepts map from. For instance, a category made up of nodes representing a collection of red items might have a limit node representing the concept red. This "red" limit node would be a generalization of the nodes which map from it. Applied to an embodied system, limits and colimits correspond to specificity and generalization in semantic space as encoded by sensory observations. Designing a categorical limit representation for the neural architecture enables generalizations to be encoded from the sensory observation space. These observation space generalization concepts correspond to semantic generalizations as mapped from observation by the particular embodied system.

This ends the Apparatus chapter, which has described computational neural modules and architectures developed for this research. Having described the elements with which experiments will be performed, the experiments themselves are described in the next chapter.

Chapter 4

Approach

This chapter describes the experimental approach used to measure the performance of the embodied neural architecture in a distal reward learning task.. Three main types of computational experiments will be performed: 1) distal reward learning, which quantifies the learned reward behavior in the virtual Morris Water Maze task, 2) persistence to goal, which., and 3) rapid transfer learning, which. Each of these are well observed behaviors in biological rats. For each type of experiment, two conditions will be compared: archirtectures with and without the categorically modified ART modules. The Results chapter contains data for each of the three experiments for the two conditions, then presents a stastical comparison of the two.

4.1 Distal Reward Learning

The first behavioral measure addresses learning with a static platform versus learning with a platform that moves randomly at each release. Moving the platform at each release creates an experimental setup where no learning is expected, or at least there can be no learning to navigate directly to the platform, an embodied entity might learn to become better at searching for novel platform positions and other such tasks that could still be learning in the always changing environment. The random

platform placement setup gives a baseline against which to compare the experimental setup where learning is expected.

Part of the experimental design is to break the releases up into blocks, where each release is the action of setting the rat in the water tank at a certain point then allowing the rat to move freely. Each block consists of four releases, one from the north release point, one from the south, one from the east, and one from the west. The four releases in a block are chosen in random order without replacement. Releasing from points in random order controls the experiment for bias that could be introduced by a constant training order.

To mimic biological experimental design, multiple experiments are performed with multiple computational rats. In biological Morris Water Maze experiments, results are collected across a sampling of rats and then statistically analyzed for evidence as to whether the sample results are likely to be indicative of behavior in the full population. If an observed behavior is not statistically valid given the sample size and variance across the sample, the results can only be said to represent behavior of those specific rats. Statistical analysis is used here in a similar manner. Without valid statistical analysis, a set of results can only provide evidence that the artificial rat model is capable of producing a some number of good results. With valid statistical analysis, it can be claimed that the sample results provide evidence that the model itself exhibits an observed behavior. In more detail; given proper statistical results, it can be stated that any size population of simulated rats produced by our model is predicted to exhibit the same behavior as the sample set of simulated rats.

In biological rats, each rat is effectively unique due to an untold number of differences in specific neural network layout and weights. These differences come about in physiological development, and due to different environments. Environmental differences are likely not large for laboratory rats, but none the less, one rat will be in the first cage in the row and another rat will be in the second cage in the row. Modern laboratory rats can be obtained with remarkable uniformity in genetically

encoded information, but no two rats are going to be unique down to the last base pair and synaptic weight. That assertion can be backed up by simply considering the average rate of genetic transcription errors and the total number of base pairs in a rat genome. There is an element of randomness in each individual rat's genome, and that would be expected to result in an element of randomness in each individual rat's neural structure. In regard to synaptic weights, two rats that were otherwise absolutely identical could catch two slightly different concentrations of the same scent one day and forever after have divergent synaptic values where the sensory activations from those scents passed through cortex.

In the simulated rats, without design otherwise, the neural structure would be identical between rats. The model parameter which is employed to model subtle differences between rats is the random search path that is taken when no expected reward stimulus is recognized. In the algorithm that governs a simulated rat's neural activations, and —through embodiment— behavior, there is a pseudo-random number generator that feeds into motor action. This pseudo-random number generator can be seeded with an initialization value. A different series of pseudo-random numbers will be generated for each seed value. Each time the same initial seed value is applied, the same series of numbers will be created. In this way a whole population of pseudo-random series is obtained that can be repeated by applying the same seed value. A different series of pseudo random numbers results in a different random search path. A different random search path results in a different instance of neural learning as different stimuli are encountered in different orders. This provides an advantage over biological experiments in that an artificial rat can be trained on a static platform location, then that rat can be compared to itself trained only with random platform locations. This is a cleaner experimental design than is possible with biology, but the nature of the statistical analysis reduces the results to a certain sized sample set with a certain variance. Being able to train the same rat, without other prior experience, with different scenarios does not actually help or hinder the

scientific validity of the comparison.

In the context of the Morris Water Maze task, and biological or social science experiments in general, the frequentest method of statistical hypothesis testing is a pervasive standard. A falsifiable null hypothesis is tested using observed data. A null hypothesis is tested by calculating how probable the observation of a set of data is if the null hypothesis were true. This probability is called a p-value. The t-test is the standard method of testing a null hypothesis between two groups. Analysis of variance (ANOVA) is the standard method for extending the t-test to more than two groups.

The null hypothesis for the first type of experiment, static platform versus random platform placement, is that over repeated releases, a rat does not learn to navigate more directly to a static platform location. In other words, the rat will not learn the location of the platform. As described above, it is not just intended to show that the computational neural model is capable of demonstrating rat like behavior but rather that any artificial rat which is an embodiment of the computational neural model is expected to demonstrate rat like behavior. Therefore a sample set of rats and an appropriate statistical method is utilized.

The R code:

```
fit <- aov(data$pl~factor(dtat$g)*factor(data$tb)
        +Error(factor(data$r)))
summary(fit)
```

is used for the analysis performed. The form of the input data is displayed in Table 4.1. In the PathLngth column, the value is the sum pathlength that the first rat took to the fixed platform location across the N, S, E, W release points. The rat number and platform behavior are indicated by the Rat 1, Group 1 indicators in that first row. There are 40 artificial rats, 20 in group 1 with the fixed platform location, and 20 in group 2 with the changing random platform location. TrlBlk indicates which trial block the pathlength measurement is from.

Rat	Group	TrlBlk	PathLngth
1	1	1	438
1	1	2	382
1	1	3	642
...			
20	1	100	384
21	2	1	302
...			
40	2	100	406

Table 4.1: Form of input data for statistical analysis in R.

4.2 Persistence to Goal

The next behavior is persistence to goal. First, a rat learns to navigate to a given platform location. Then that platform is removed. Persistence to goal behavior is observed when the rat swims around the previous location of the platform, even though the platform is no longer present.

Persistence to goal behavior is tested by training an artificial rat on a fixed platform location. Training occurs by randomly selecting release points from the four release locations without replacement, just as in the basic task, so that each trial block includes having the rat trained once from each release point. Three more blocks of releases are then performed with no platform present in the tank. The first three blocks from training are used as the untrained performance measure. These first three blocks show the rat still learning where the fixed platform is located. The untrained rat should have no preference for persisting around the platform location because the untrained rat has no knowledge of the platform location. The three untrained blocks are compared against three blocks of the trained rat with no platform in the tank. Therefore, the performance as measured by percentage of time spent within the platform region of a rat with no training is compared to the performance as measured by percentage of time within the platform region of a rat that learned to navigate to an escape platform. For the ANOVA, the three untrained blocks are group one and the three trained blocks are group two.

The null hypothesis is that a trained rat will show no preference toward the learned platform location compared to a rat which has not been trained on the platform location. Preference toward the learned platform location is measured as the percentage of release time that the rat spends within a given distance of the location where the platform is or was during training. Percentages of release time in proximity to the platform is calculated across the three blocks for each of the twenty experiment rats for each of the two groups. This gives a vector of forty measurements of observed persistence. The forty measurements are divided into two groups. Group one is the persistence of untrained rats. Group two is the persistence of trained rats with the platform removed. The p-value resulting from an analysis of variance of the results gives the probability that the analyzed set of results could have been generated if the null hypothesis were true.

4.3 Rapid Transfer Learning

The final empirically measured behavior is known as rapid learning or transfer learning. The rat learns to escape the water maze from one release point, and then is released from a different point. The behavior being tested is whether the trained rat will find the platform any quicker from the novel release point than a rat which has never performed the Morris Water Maze task at all.

There are two groups of rats, those which are have never been released into a tank, and those which have been released and learned to find the platform from one release point. To account for possible bias from certain release point training-testing combinations, all possible combinations are tested. I.e. a rat will learn to navigate to the platform, with no prior experience, from each of the north, south, east, and west release points. Taking advantage of the ability to have multiple copies of the same rat, a rat which has only learned the north point is released from each of the east, south, and west points to test transfer learning from north to east, north to south and north

to west respectively. Likewise let the rat learn to escape from the east release point, then release from the north, south, and west. Learn from south, then release from west, north, and east. Learn from west, then release from north, east, and south. In all cases pathlength to goal is measured. The first three blocks of learning are recorded for the rats which have never been released in any experiment. These three naive blocks give a baseline pathlength for unexperienced rats from the north, south, east, and west. Comparison is made of the pathlength to goal for an unexperienced rat released from the north to the pathlength for a rat which has learned to navigate to goal from the other three directions, thereby getting three comparisons for each initial release point. That makes four unexperienced measurements, each of which is compared to three measurements from rats with prior experience for a total of twelve comparisons per rat. This is repeated for a set of twenty sample computational rats.

Therefore, the performance as measured by pathlength of a rat with no training from any release point is compared to the performance as measured by pathlength of a rat with training from the wrong release point. By "training from the wrong release point" it is meant that the rat has been trained from a different release point to find the platform in the current tank, however the rat has not been trained from the current release point therefore the rat only has learning from a "wrong" release point and none from the tested release point. For the ANOVA, the three untrained blocks are group one and the three trained blocks are group two.

The term "wrong" is used to unambiguously connote that transfer learning is occurring from training on one release point to performance on learning to navigate from a different release point. A positive result will be if this wrong training improves performance.

This protocol results in sixteen subgroups. Four subgroups are the four cardinal training release points. Then there are an additional four sets of three subgroups for the three transfer test release points following each training release point. The pathlength measures of the first three trial blocks of the four training release point

subgroups forms group one. The pathlength measures of the first three trial blocks for the twelve transfer subgroups forms groups two. These group one and two are the same as described in the previous paragraph, just now explained in more detail.

The null hypothesis to test transfer learning is that learning to navigate to reward from one release point will not improve the ability to navigate to reward from another release point. For the analysis of variance, group one is comprised of the trials with no prior experience. Pathlength is the dependent variable. Group and trial number are the independent variables.

4.4 Comparison of Baseline to Limits Architectures

The first two sections of the Results chapter will give results from the statistical analysis of observed behavior from the baseline architecture and the categorical limit architecture considered separately. The third section of the Results will present a statistical analysis of the comparison of the two.

To analyze the baseline versus the limit architectures, the stationary platform group from the baseline architecture analysis is labeled as group one for each experiment. The stationary platform group from the categorical limit architecture is labeled as group two. This new grouping compares the learning group from each architecture. For each experiment type, the null hypothesis is that the group using the categorical limits architecture will show no significant difference from the group using the baseline architecture.

The treatment group from the Distal Reward Learning task was the group with fixed platform location. Having a fixed platform, the treatment group had a chance to exhibit the behavior of learning to repeatedly navigate to the platform. The other group, with a platform location that changed at each release, never had a chance to perform the learning behavior of interest. In the Persistence to Goal experiment, the

treatment group was the group which learned to navigate to the platform, then had the platform removed. In the Rapid Transfer Learning experiment, the treatment group was the group which first learned to navigate to the platform from one release point, then was released from a novel point.

Chapter 5

Results

This Results chapter presents data from representative samples from the set of all experiments, and then statistical results for all experiments described in the Approach chapter. In general a representative sample is to displayed from behavior data generated in the simulation of one of the twenty computational rats experimented upon. Recall that each of the twenty rats is uniquely simulated based on its random number generation seed value. This seed value determines the random path that will be taken in searching for the escape platform. Past the representative graphs, data for many more individual rat experiments can be found in Appendix F.

Table 5.1 provides release coordinates in the simulated environment for the four release point and for the escape platform. Direct distances from release point to platform are calculated for reference when considering learned pathlengths in the results data. Note that for any experiment with a fixed platform location, the platform is positioned at coordinates $(100, 66)$. This location was chosen as being similar to placement used in biological Morris water maze experiments. The coordinate system origin is at the center of the tank. The platform is a 30 square with side length 30, the tank has diameter 500, and the rat has diameter 20. Rate movement speed is 4 spatial units per timestep and pathlength is recorded in movement steps. As such, distance to platform is normalized to movement step for comparison with path-

Release Point	X Coord.	Y Coord.	Distance to Platform	Normalized Distance to Platform
N	0	200	136	34
S	0	-200	253	62
E	200	0	89	22
W	-200	0	276	69

Table 5.1: Release point coordinates and platform distance from release points. Normalized distance is for comparison to data below.

Parameter	Architecture Module	Value
ρ	Hippocampal LAPART	0.94
ρ	Sensory or Motor ART	0.999
Decay Constant	Sensory or Motor TInt	0.65

Table 5.2: Architecture parameters used to produce reported results

length data. Distance to platform also subtracts off the rat radius and distance from platform center to corner such that an optimal pathlength is calculated.

Orientation upon release is toward the center of the tank. In biological experiments, rats are often oriented toward the wall upon release, however this orientation is used to prevent the rat from "cheating" and beginning to observe its location before release. The computational rats will never begin taking sensory input until instructed to do so, therefore there is no need to orient toward the wall.

Further implementation parameters which are necessary for reproduction of these results are the vigilance parameter ρ of the ART modules in the neural architecture and the decay constant of the temporal integrator modules. These parameters are reported in Table 5.2. These values are also available in the .spec file which provides the specifications to eLoom for a simulation run. These values were arrived at through iterative computational experimentation.

5.1 Baseline Architecture

5.1.1 Distal Reward Learning

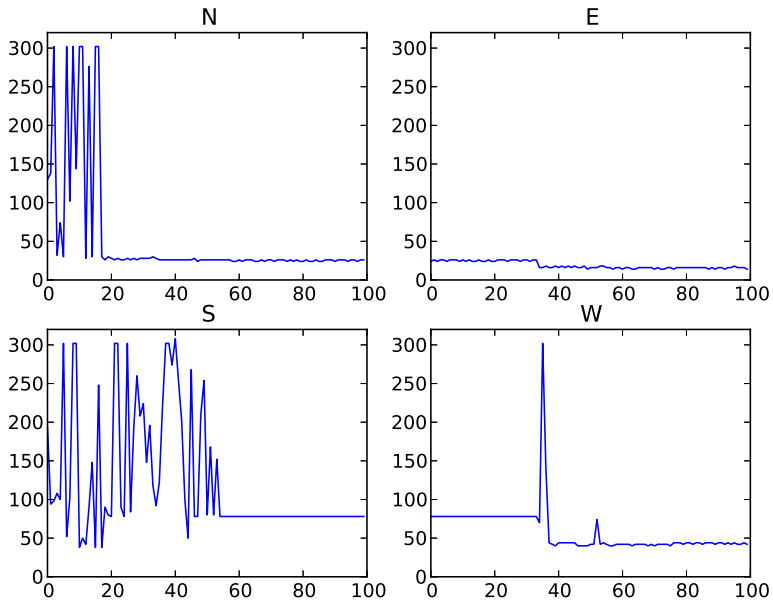

Figure 5.1: Graph of pathlength for each release for one experimental trial. Releases cycle randomly between north, south, east, west release points chosen without replacement until all four locations have been chosen at which point all are place back into the pool for the next selection. The graph shows behavior measuring data before any numerical analysis.

Figure 5.1 shows a graph of pathlength vs. release number for each of the four release points. Figure 5.2 shows sum pathlength by block. Recall that each block is a random selection of release points chosen without replacement from north, south, east, and west.

The progression of pathlengths from rat release to trial end given platform placement at a new random location for each release is seen in Figure 5.3. Recall that a trial can end either because the rat encounters the platform or because the rat has

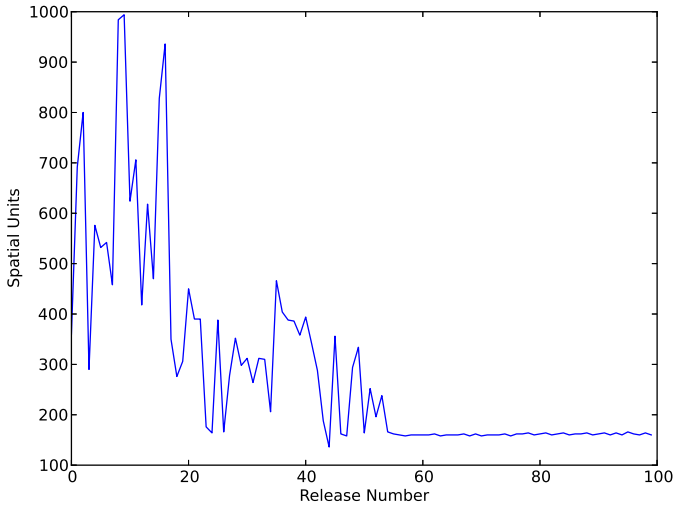

Figure 5.2: Graph of sum pathlength for each block of releases. At the N^{th} block, the rat will have been release from each point N times. The graph illustrates learning averaged over all four release points.

been searching past a given amount of time. In biological rats, the time cutoff is to prevent exhaustion and possible drowning. The cutoff is simulated at a pathlength of 300 spatial units, which is equivalent to a fixed time as the simulated rat moved at a constant speed. Block pathlengths for the same experiment are seen in Figure 5.4.

The simulated Morris water maze experiment was conducted for each of the population of 20 computational rats as described in the preceding chapter. A one-way between subjects ANOVA was conducted to compare the effect of repeated training trials on pathlength to reward in stationary platform and random location platform conditions. Results were $F(2, 38) = 17.002, p = 0.0001955, MeanSquared = 4089219$, $\eta^2 = 0.55$.

Table 5.3 shows the sum pathlength over the last half of each experimental trial for each computational rat. The last half pathlengths are summed to show the effect of learning. The summation span of the last half was chosen arbitrarily, the last third or last quarter would provide similar analytic result. While there is no explicit

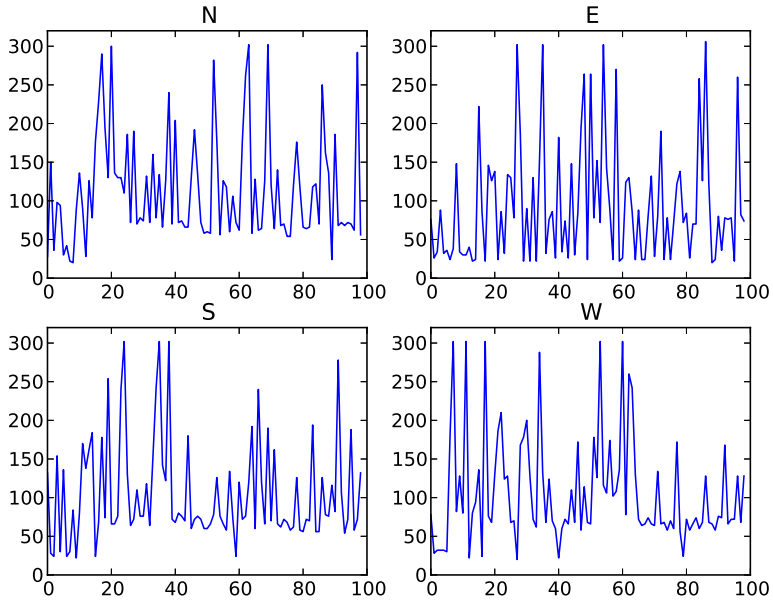

Figure 5.3: Graph of pathlength for each release with random platform location per release.

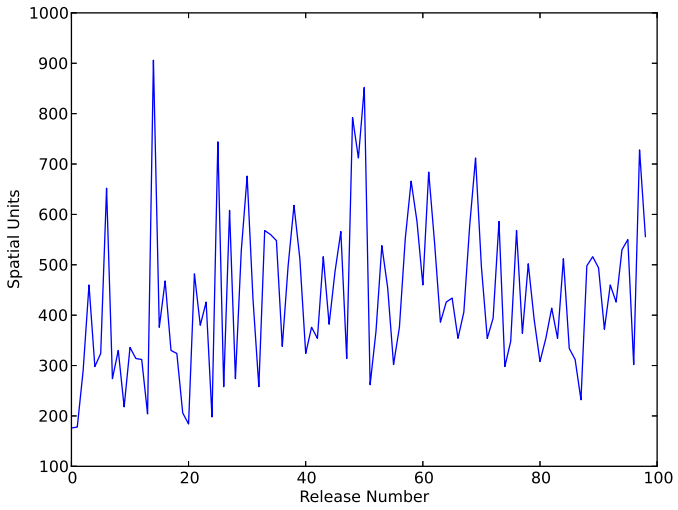

Figure 5.4: Graph of sum pathlength for each block of releases with random platform location per release.

Rat	Random Platform	Fixed Platform
0	21268	11986
1	18418	14872
2	23116	16626
3	17946	15496
4	23206	11218
5	20732	8266
6	18764	14590
7	20244	11906
8	20132	17560
9	21894	24598
10	21372	9766
11	20362	13720
12	17642	19078
13	23352	18848
14	18012	16436
15	19668	20270
16	20576	12478
17	19218	17484
18	17770	6356
19	21666	13924
Average	20268	14774
SD	1837	4339

Table 5.3: Comparison of sum pathlength for the last half of each experiment trial for each computational rat, showing random platform sum pathlength vs. fixed platform sum plathlength.

behavioral threshold between the first half of releases in a trial and the second half, it is generally true that the last half behavior will demonstrate more learning than the first half. Therefore it is expected to see the greatest effect of learning in the last half behavior comparing random platform location versus a fixed platform location.

5.1.2 Persistence to Goal

The following graphs show representative data from computational rats performing the persistence to goal task. The first graph shows data from an untrained rat that has not yet learned to navigate to the platform. The second graph shows data from

a trained rat that has learned to navigate to the platform, however the platform has been removed so the rat continues to search until the release time limit is reached.

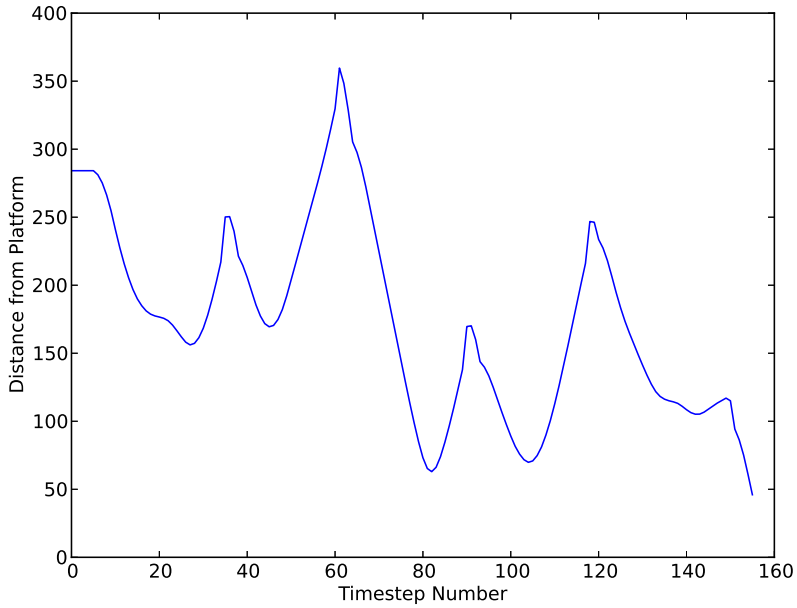

Figure 5.5: Distance from platform location for untrained rat number 0.

Figure 5.5 shows a graph of distance from platform vs. time as the untrained rat is attempting to find that platform. Figure 5.6 shows distance from the platform for the same rat after training, but with no platform actually present in the tank. By removing the platform, measurements can be taken as to how much time a rat spends within a threshold distance of 60 spatial units from the center of the platform. Recall that the platform is a 30 spatial unit square and the tank is 500 spatial units in diameter. Given the area of platform relative to the tank, a naive estimate of time spent near the platform for a randomly searching rat is 5.76%.

Table 5.4 shows the data used for the persistence to goal analysis. As described in the Approach chapter, there are two groups of 20 computational rats each. Group one is made up of trained rats and group two is untrained rats.

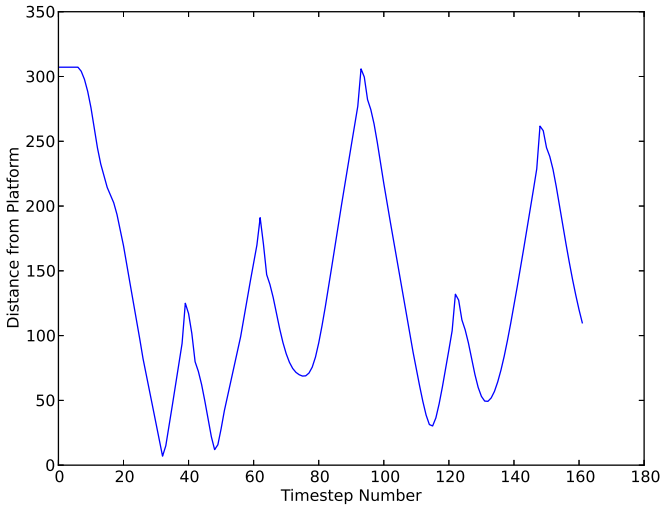

Figure 5.6: Distance from previous platform location for trained rat number 0 with no platform in tank.

The persistence to goal experiment was conducted using each of the population of 20 computational rats as described in the Approach chapter. A one-way between subjects ANOVA was conducted to compare percent of time spent within learned platform region in trained and untrained agent conditions. Results were $F(1, 38) = 156.49, p = 4.79 \times 10^{-15}, MeanSquared = 4.6172, \eta^2 = 0.56$.

5.1.3 Rapid Transfer Learning

The following graphs show representative data from the rapid transfer learning experiments. The graphs display pathlength traveled per release. A release ends either when the rat finds the platform or the time limit is reached. The first graph is of a computational rat being trained from the west release point. The second graph is of the same rat being tested from the west release point after having been trained on the north release point. Recall that the rapid transfer task is to, after having been trained to navigate to the platform from a given release point, to then navigate to the platform from a novel release point. The two graphs illustrate the transfered

Rat	Percent In Threshold	
	Untrained	Trained
0	5.60	6.42
1	5.41	6.20
2	5.23	6.14
3	5.60	5.93
4	5.73	6.07
5	5.65	6.07
6	5.66	6.35
7	5.17	6.05
8	5.39	6.18
9	5.26	6.41
10	5.21	6.24
11	5.70	6.00
12	5.73	6.26
13	5.63	6.05
14	5.57	6.25
15	5.64	6.23
16	5.61	6.29
17	5.54	5.99
18	5.26	6.26
19	5.65	6.44

Table 5.4: Percent time spent within a threshold distance of the reward platform for each computational rat, showing trained vs. untrained performance.

benefit that occurs when learning to navigate from the west release point in two conditions. The first condition is an unexperienced condition with no prior training on the navigation task in any form. The second condition is an experienced condition with prior training on the navigation task in a different form (i.e. from a different release point).

Figure 5.7 shows a graph of pathlength from release to trial end as the untrained rat is attempting to find the platform after being released from the north release point. Note that in contrast to the example graph given earlier, this example shows a rat which is unable to successfully learn navigation over the course of the experiment. Figure 5.8 shows pathlength per release for the same rat after having trained on the north release point then in this experiment attempting to learn from the west release

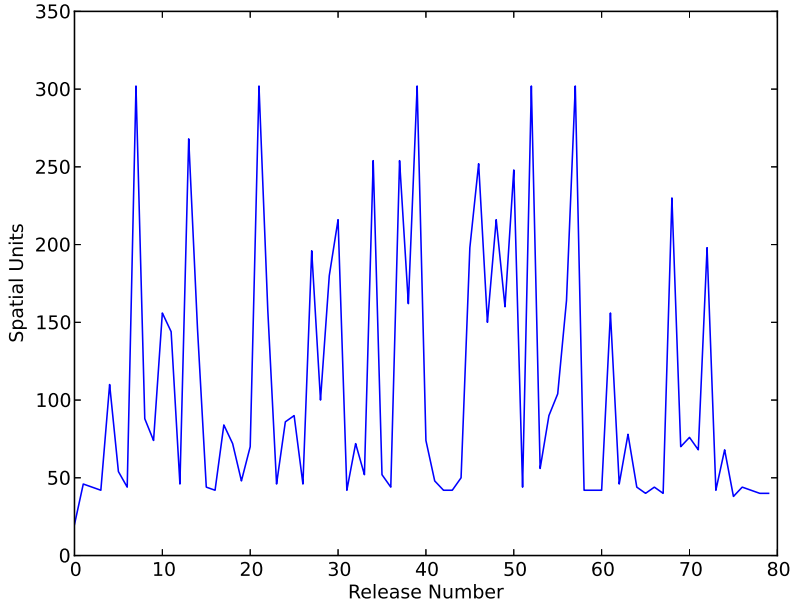

Figure 5.7: Pathlengths of releases during training for rat 0 from north release point.

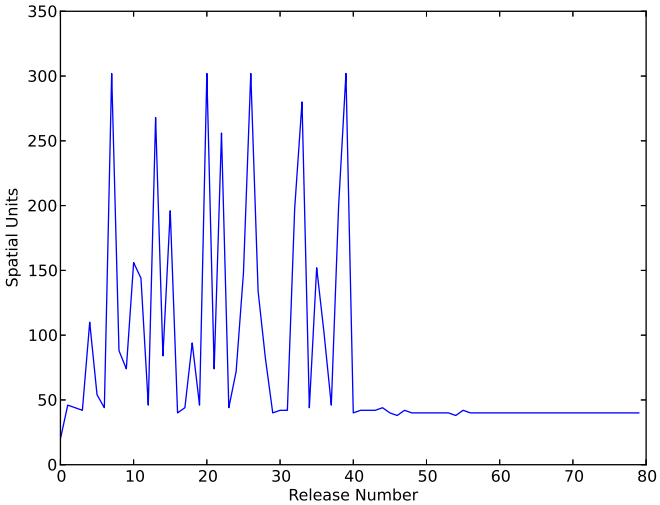

Figure 5.8: Pathlengths of releases during testing for rat 0 trained from north release point followed by the west release point.

Release	Average Pathlength	Standard Deviation
N	5786	2295
EN	5892	2115
SN	5075	2064
WN	5708	2528
E	2160	2174
NE	1481	1965
SE	1219	798
WE	1195	637
S	8102	2000
NS	6834	2571
ES	8290	2015
WS	6485	2540
W	7903	1368
NW	7266	1838
EW	7843	1443
SW	7019	1715

Table 5.5: Average pathlengths with standard deviation across rats for each release and pair of releases.

point.

Table 5.5 shows average and standard deviation across all rats for the set of pathlengths from each release or pair of releases. Single releases are without training. Paired releases are without training before the pair, then training occurs from the first release point, and then pathlengths are measures for the second release point.

The transfer learning experiment was conducted exhaustively for all combinations of initial training release point and subsequent testing release point for each of the population of 20 computational rats as described in the preceding chapter. A one-way between subjects ANOVA was conducted to compare the effect of repeated training trials on pathlength to reward in experienced and unexperienced agent conditions. Results were $F(1, 38) = 4.5102, p = 0.04026, MeanSquared = 2372651$, $\eta^2 = 0.72$.

5.2 Categorical Limits Architecture

5.2.1 Distal Reward Learning

The following graphs show pathlengths from release to reward for an example computational rat with the categorical limits augmentation. Random platform location results are taken to be the same as for the baseline architecture under the assumption that both architectures use the same default navigation (random search) behavior. If no learning occurs for the random platform case, then the learning portion of the architecture is irrelevant and it is only the learning portion of the architectures which differ between baseline and categorical limit types.

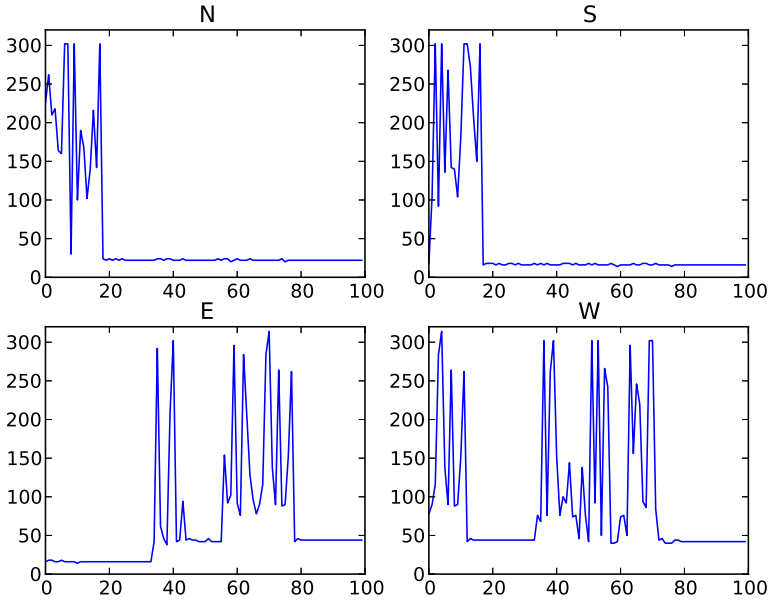

Figure 5.9: Graph of pathlength for each release for computational rat 4 learning to navigate to a fixed platform.

Figure 5.9 shows a graph of pathlength vs. release number for each of the four release points. Figure 5.10 shows sum pathlength by block. Recall that each block is

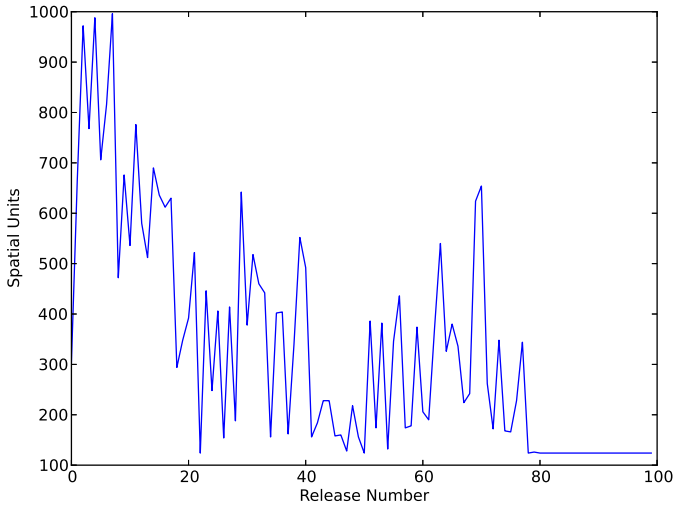

Figure 5.10: Graph of sum pathlength for each block of releases for computational rat 4. At the N^{th} block, the rat will have been release from each point N times. The graph illustrates learning averaged over all four release points.

a random selection of release points chosen without replacement from north, south, east, and west.

The simulated Morris water maze experiment was conducted for each of the population of 20 computational rats. A one-way between subjects ANOVA was conducted to compare the effect of repeated training trials on pathlength to reward in stationary platform and random location platform conditions. Results were $F(2, 38) = 5.2639, p = 0.02739,$ Mean Squared $= 1788867$, $\eta^2 = 0.53$.

Table 5.6 shows the sum pathlength over the last half of each experimental trial for each computational rat. As was done for the baseline architecture, the last half pathlengths are summed to show the effect of learning.

5.2.2 Persistence to Goal

The following graphs show representative data from computational rats with the categorical limit augmentation performing the persistence to goal task. The first

Rat	Random Platform	Fixed Platform
0	21268	22688
1	18418	15730
2	23116	6024
3	17946	18862
4	23206	6444
5	20732	8000
6	18764	14546
7	20244	14866
8	20132	16745
9	21894	23642
10	21372	13974
11	20362	15364
12	17642	12616
13	23352	14218
14	18012	13222
15	19668	9362
16	20576	12982
17	19218	11522
18	17770	6162
19	21666	16086
Average	20268	13653
SD	1838	4904

Table 5.6: Comparison of sum pathlength for the last half of each experiment trial for each computational rat, showing random platform sum pathlength vs. fixed platform sum plathlength for the categorical limits augmented architecture.

graph shows data from an untrained rat that has not yet learned to navigate to the platform. The second graph shows data from a trained rat that has learned to navigate to the platform, however the platform has been removed so the rat continues to search until the release time limit is reached.

Figure 5.11 shows a graph of distance from platform vs. time as the untrained rat is attempting to find that platform. Figure 5.12 shows distance from the platform for the same rat after training, but with no platform actually present in the tank. By removing the platform, measurements can be taken as to how much time a rat spends within a threshold distance of 60 spatial units from the center of the platform. Recall that the platform is a 30 spatial unit square and the tank is 500 spatial units

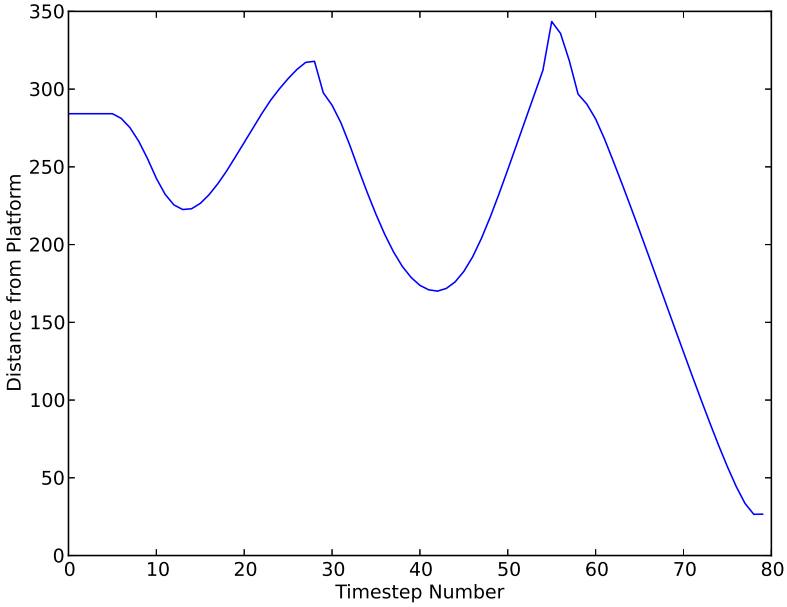

Figure 5.11: Distance from platform location for untrained rat number 0.

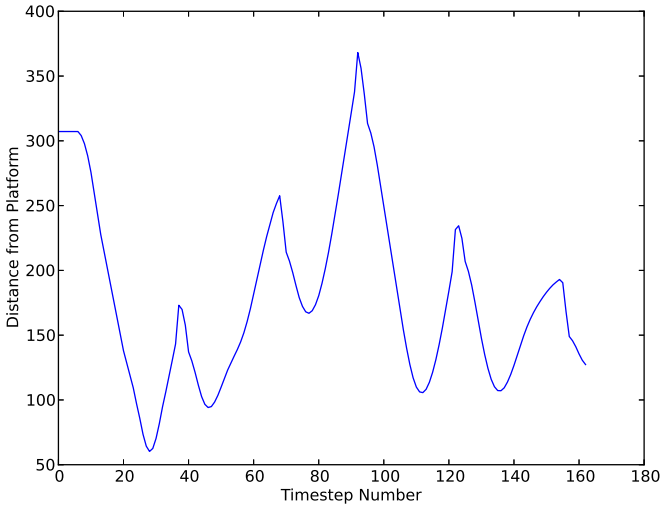

Figure 5.12: Distance from previous platform location for trained rat number 0 with no platform in tank.

Rat	Percent In Threshold	
	Untrained	Trained
1	5.62	6.26
2	5.41	6.34
3	5.23	6.25
4	5.6	6.39
5	5.73	6.41
6	5.65	6.38
7	5.62	6.27
8	5.17	6.16
9	5.39	6.03
10	5.26	6.57
11	5.2	6.56
12	5.63	6.29
13	5.73	6.3
14	5.63	6.32
15	5.11	6.35
16	5.64	6.26
17	5.58	6.29
18	5.53	6.41
19	5.26	6.34
20	5.65	6.6

Table 5.7: Percent time spent within a threshold distance of the reward platform for each computational rat, showing trained vs. untrained performance.

in diameter.

Table 5.7 shows the data used for the persistence to goal analysis. As described in the Approach chapter, there are two groups of 20 computational rats each. Group one is made up of trained rats and group two is untrained rats.

The persistence to goal experiment was conducted using each of the population of 20 computational rats as described in the previous chapter. A one-way between subjects ANOVA was conducted to compare percent of time spent within learned platform region in trained and untrained agent conditions. Results were $F(1, 38) = 243.26, p = 2.2 \times 10^{-16}, MeanSquared = 7.3445$, $\eta^2 = 0.57$.

5.2.3 Rapid Transfer Learning

The following graphs show representative data from the rapid transfer learning experiments performed by computational rats with the categorical limits augmentation. The graphs display pathlength traveled per release. A release ends either when the rat finds the platform or the time limit is reached. The first graph is of a computational rat being trained from the west release point. The second graph is of the same rat being tested from the west release point after having been trained on the north release point. Recall that the rapid transfer task is to, after having been trained to navigate to the platform from a given release point, to then navigate to the platform from a novel release point. The two graphs illustrate the transfered benefit that occurs when learning to navigate from the west release point in two conditions. The first condition is an unexperienced condition with no prior training on the navigation task in any form. The second condition is an experienced condition with prior training on the navigation task in a different form (i.e. from a different release point).

Figure 5.13 shows a graph of pathlength from release to trial end as the untrained rat is attempting to find the platform after being released from the north release point. Note that in contrast to the example graph given earlier, this example shows a rat which is unable to successfully learn navigation over the course of the experiment. Figure 5.14 shows pathlength per release for the same rat after having trained on the north release point then in this experiment attempting to learn from the west release point.

Table 5.8 shows average and standard deviation across all rats for the set of pathlengths from each release or pair of releases. Single releases are without training. Paired releases are without training before the pair, then training occurs from the first release point, and then pathlengths are measures for the second release point.

The transfer learning experiment was conducted exhaustively for all combinations of initial training release point and subsequent testing release point for each of the

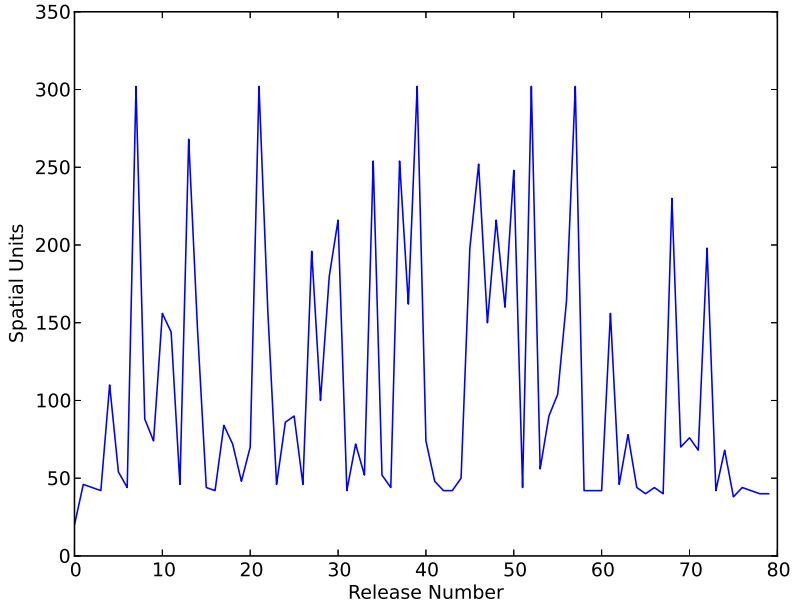

Figure 5.13: Pathlengths of releases during training for rat 0 from the west release point.

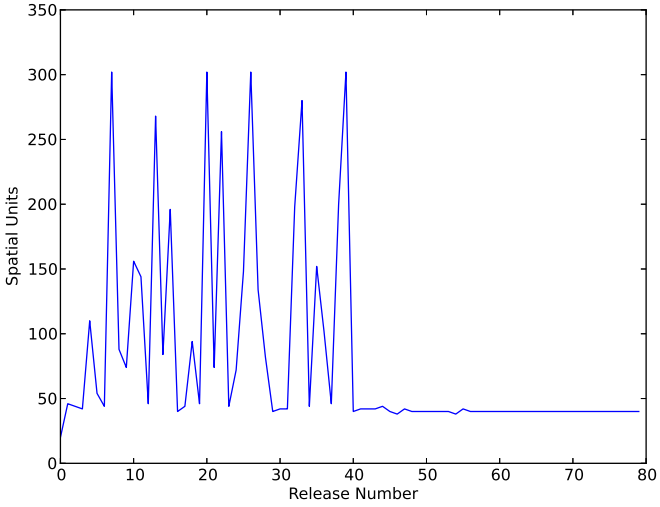

Figure 5.14: Pathlengths of releases during testing for rat 0 trained from the north release point followed by the west release point.

Release	Average Pathlength	Standard Deviation
N	5882	2393
EN	5819	2174
SN	5492	2584
WN	4597	2327
E	2224	2302
NE	1486	1987
SE	1219	798
WE	1195	637
S	8124	2119
NS	7033	2702
ES	8453	2185
WS	6653	2513
W	7903	1368
NW	6914	2551
EW	6759	2622
SW	7182	1446

Table 5.8: Average pathlengths and standard deviation across rats for each release and pair of releases.

population of 20 computational rats as described in the preceding chapter. A one-way between subjects ANOVA was conducted to compare the effect of repeated training trials on pathlength to reward in experienced and unexperienced agent conditions. Results were $F(1, 38) = 6.89, p = 0.01240, MeanSquared = 38354414, \eta^2 = 0.71$.

5.3 Comparison of Baseline to Limits Architectures

5.3.1 Distal Reward Learning

Table 5.9 shows sum last half pathlength of the baseline and categorical limits architectures. The data for each architecture represents the fixed platform case, where learning can occur. As such, these results show the learning performance difference between the architectures.

Rat	Baseline	Limits
0	11986	22688
1	14872	15730
2	16626	6024
3	15496	18862
4	11218	6444
5	8266	8000
6	14590	14546
7	11906	14866
8	17560	16745
9	24598	23642
10	9766	13974
11	13720	15364
12	19078	12616
13	18848	14218
14	16436	13222
15	20270	9362
16	12478	12982
17	17484	11522
18	6356	6162
19	13924	16086
Average	14774	13653
SD	4339	4904

Table 5.9: Comparison of sum pathlength for the last half of each experiment trial for each computational rat, showing fixed platform performance for each architecture.

A one-way between subjects ANOVA was conducted to compare the effect of the baseline architecture to the categorical architecture on each of the three Morris water maze tasks. Results of the ANOVA for the basic Morris water maze task treatment groups in the non-categorical-limits augmentation versus the categorical limits augmentation case were $F(1, 38) = 1.0196, p = 0.319, MeanSquared = 468809$, $\eta^2 = 0.48$.

5.3.2 Persistence to Goal

Table 5.10 shows the data used for the persistence to goal analysis. As described in the Approach chapter, there are two groups of 20 computational rats each. Group one

Rat	Percent In Threshold	
	Baseline	Limits
1	6.2602	6.4200
2	6.3436	6.2012
3	6.2483	6.1420
4	6.3911	5.9278
5	6.41	6.0653
6	6.3758	6.0745
7	6.2666	6.3509
8	6.1625	6.0455
9	6.0331	6.1788
10	6.5662	6.4098
11	6.5589	6.2417
12	6.2931	5.9966
13	6.2979	6.2630
14	6.3224	6.0493
15	6.3543	6.2483
16	6.2577	6.2268
17	6.2882	6.2883
18	6.4072	5.9900
19	6.3371	6.2597
20	6.6031	6.4367

Table 5.10: Percent time spent in vicinity of reward platform for each computational rat, showing baseline vs. categorical limits architecture performance.

is made up of trained rats and group two is untrained rats. Results of the ANOVA for the persistence to goal task were $F(1, 38) = 10.730, p = 0.002254, MeanSquared = 0.21960$, $\eta^2 = 0.49$.

5.3.3 Rapid Transfer Learning

Table 5.11 shows the difference between average pathlengths between the baseline and categorical limits architecture. ANOVA results for the rapid transfer learning task for 20 rats using the baseline architecture compared to those 20 rats using the categorical limits architecture were $F(1, 285) = 8.6911$, $p = 0.003462$, $MeanSquared = 38354414$, $\eta^2 = 0.5$.

Release	Baseline	Limits
N	5786	5882
EN	5892	5819
SN	5075	5492
WN	5708	4597
E	2160	2224
NE	1481	1486
SE	1219	1219
WE	1195	1195
S	8102	8124
NS	6834	7033
ES	8290	8453
WS	6485	6653
W	7903	7903
NW	7266	6914
EW	7843	6759
SW	7019	7182

Table 5.11: Average pathlenths across all rats for each release condition for the baseline and the categorical limits architecture.

		Architecture		
		Baseline	Limits	Baseline vs. Limits
Experiment	Distal Reward	0.0001955	0.02739	0.319
	Persistence to Goal	4.79×10^{-15}	2.2×10^{-16}	0.002254
	Rapid Transfer Learning	0.04026	0.01240	0.003462

Table 5.12: Summary of p-value for statistical analysis of computational experiment results

5.4 Summary

This section provides Table 5.12, a summary table of results. The summary table lists p-value for each experiment for each architecture.

This Results chapter has presented data and statistical analysis of embodied architectures. Having now shown the quantitative measures, the next chapter will give insight into implications of these measures.

Chapter 6

Discussion

In this chapter, the empirical analysis of the behavior of the embodied neural architecture is discussed. First, statistical analysis of the basic embodied architecture performing distal reward, persistence to goal, and rapid transfer learning tasks is discussed. Then the statistical analysis of the categorical limits supporting embodied architecture during the same task performance is considered. Following that, the original architecture is compared to the limits architecture. Finally, results are discussed in the context of behavior observed in biological rats.

6.1 Statistical Analysis of Baseline Architecture

Statistical Analysis is employed to provide evidence that observed behavior is not a product of chance. Rather, the neural architecture itself is showing significant learning of distal reward behavior.

6.1.1 Distal Reward Learning

Using analysis of variance (ANOVA) to test the statistical difference between the data for the individual release pathlength for the individual rats between the static platform group and the random platform location group yields a p-value of 0.0001955,

indicating that there is a less than a 0.02 percent chance that the observed results were observed in a system where the null hypothesis is valid. A p-value of 0.05 is generally accepted as the cut off for statistical significance [106]. The null hypothesis was that learning to a fixed platform location would be performed no better than learning to a random platform location. Since there is effectively no learning to a random platform location, were the null hypothesis true, it would mean that no learning had occurred in the experiment. Table 5.3 in Section 5.1.1 shows sum pathlenth over the last half of releases of the experiment for each rat. It is seen that for most rats, sum pathlength for the releases making up the last half of each trial is lower in the fixed platform case than the random platform case. The average over all subjects was 14774 with fixed platform and 20268 with random platform. Shorter pathlengths to the fixed platform indicate that the platform location was learned. Based on a significant result against the null hypothesis, along with the data in Table 5.3 which shows evidence of learning, it is concluded that embodiments of the computational model will learn to navigate to a fixed platform.

6.1.2 Persistence to Goal

As shown in results section 5.1.2, the average percent time spent near reward for untrained rats is 5.51. The average percent time spent near reward for trained rats is 6.19. A p-value of $4.79x10^{-15}$ is observed for ANOVA between the two groups. This indicates that for the two groups of twenty simulated rats observed, there is a probability of $4.79x10^{-15}$ that the behavioral difference, as measured between the two groups, could have occurred given a true null hypothesis. In this case, the null hypothesis was that trained rats will spend an equal amount of time searching around the location of reward in the water maze as will untrained rats. Therefore the trained rats did, with statistical significance, spend more time swimming within the platform region.

To see that the p-value makes sense, observe the underlying data displayed in

	Initial	N	E	S	W
N	5786		5892	5075	5708
E	2160	1481		1219	1195
S	8102	6834	8290		6485
W	7903	7266	7843	7019	
	Differences				
N			-106	711	78
E		679		941	965
S		1268	-188		1618
W		637	60	884	

Table 6.1: Differences of pathlenths between naive release and releases with potential transfer learning.

Table 5.10. PctInThresh records the percent of total tank time which the rat spent within a constant distance from the center of the platform location. It can be observed that there is no overlap between the fractions of time which the two population samples spent in the platform region. On an absolute scale, there is not a large magnitude in the difference between measured group behaviors, but the analysis of variance allows us to none the less assign meaningful statistical significance to the measures.

Therefore, it can be said that the embodied computational neural architecture is capable of performing the persistence to goal behavior. This same persistence to goal is observed in biological rats, as will be discussed in a later section.

6.1.3 Rapid Transfer Learning

Results section 5.1.3 reports the average across rats of the sum across releases for each trial of the experiment. The top half of table 6.1 shows the same values in a new format. The format of this table shows naive release points on the row headers and potential transfer learning release points on the column headers. The bottom half shows the difference in pathlength between naive releases and releases with the potential for transfer learning. Naive releases occur with the computational rat having not prior experience. Releases with the potential for transfer learning occur

when a rat has previously been trained from a different release point. Across potential transfer learning releases from the north release point, there is an average pathlength difference of 228 from the naive values minus the potential transfer learning values. For the east release point, the average difference was 862. The south difference was 899. The west difference was 527. These differences indicate that on average, transfer learning occurred as a positive difference indicates that the transfer learning average pathlengths were lower. A p-value of 0.04026 for the observed rapid transfer learning behavior. Recall that the null hypothesis for rapid transfer learning is that given a rat will be released from point X, a rat trained from some release point other than X will show no increased performance as compared to a rat with no morris water tank training at all. 0.04026 is the weakest p-value observed for the three experiments. None the less, it is low enough to be considered statistically significant by most criteria. The embodied neural architecture therefore demonstrates reasonably sound rapid transfer learning behavior.

6.2 Statistical Analysis of Categorical Limit Architecture

The categorical limit architecture is considered both by itself and in the context of the architecture that does not implement categorical limits. The limit architecture alone is discussed in this section. The limit architecture relative to the non-limit architecture is discussed in the next section.

6.2.1 Distal Reward Learning

Using analysis of variance (ANOVA) to test the statistical difference between the data for the individual release pathlength for the individual rats between the static platform group and the random platform location group yields a p-value of 0.02739,

indicating that there is a less than a 0.02 percent chance that the observed results were observed in a system where the null hypothesis is valid. The null hypothesis was that learning to a fixed platform location would be performed no better than learning to a random platform location. Since there is effectively no learning to a random platform location, were the null hypothesis true, it would mean that no learning had occurred in the experiment. Table 5.6 in Section 5.2.1 shows sum pathlenth over the last half of releases of the experiment for each rat. It is seen that for most rats, sum pathlength for the releases making up the last half of each trial is lower in the fixed platform case than the random platform case. The average over all subjects was 13653 with fixed platform and 20268 with random platform. Shorter pathlengths to the fixed platform indicate that the platform location was learned. Based on a significant result against the null hypothesis, along with the data in Table 5.6 which shows evidence of learning, it is concluded that the categorical limits augmented architecture is capable of learning to navigate to reward in the simulated Morris water maze task.

6.2.2 Persistence to Goal

Results section 5.2.2 gives results showing statistical significance for the limits architecture performing the persistence to goal task. The average percent time spent near reward for untrained rats is 5.48. The average percent time spent near reward for trained rats is 6.34. The p-value was 2.2×10^{-16}. The null hypothesis was that the untrained rats would spend an equal amount of time near the reward as would the trained rats. As significant evidence was observed against the null hypothesis, it is concluded that there was an effect given rat training and it can be observed that the trained rats persisted in navigation to and around the location at which they learned the reward platform to be found.

	Initial	N	E	S	W
N	5882		5819	5492	4597
E	2224	1486		1219	1195
S	8124	7033	8453		6653
W	7903	6914	6759	7182	
	Differences				
N			63	390	1285
E		738		1005	1029
S		1091	-329		1471
W		989	1143	721	

Table 6.2: Differences of pathlenths between naive release and releases with potential transfer learning.

6.2.3 Rapid Transfer Learning

Results section 5.1.3 reports the average across rats of the sum across releases for each trial of the experiment. The top half of table 6.1 shows the same values in a new format. The format of this table shows naive release points on the row headers and potential transfer learning release points on the column headers. The bottom half shows the difference in pathlength between naive releases and releases with the potential for transfer learning. Naive releases occur with the computational rat having not prior experience. Releases with the potential for transfer learning occur when a rat has previously been trained from a different release point. Across potential transfer learning releases from the north release point, there is an average pathlength difference of 579 from the naive values minus the potential transfer learning values. For the east release point, the average difference was 924. The south difference was 744. The west difference was 951. These differences indicate that on average, transfer learning occurred as a positive difference indicates that the transfer learning average pathlengths were lower. With a p-value of 0.0124, the categorical imits implementing embodied architecture performs rapid transfer learning with statistical significance. The results are reported in section 5.2.3.

6.3 Comparison of Baseline Architecture to Categorical Limit Architecture

The prior two sections discussed the results of statistically analyzing observed behavior for the baseline architecture and the categorical limit architecture. That is, each architecture was analyzed by itself. This section discusses the original architecture compared to the categorical limits architecture.

6.3.1 Distal Reward Learning

For the baseline architecture, the average over all subjects for sum pathlength over experiment releases of the baseline architecture was 14774. For the categorical limits architecture, the average over all subjects for sum pathlength over experiment releases of the limits architecture was 13653. As reported in results section 5.3, p-value for ANOVA of the baseline vs. limits architectures was 0.319. Even though the average second half pathlength is shorter for the limits architecture, we don't conclude that the ability to represent categorical limits makes a difference on the basic distal reward learning task because the ANOVA p-value indicates too much variance in the individual results to allow a conclusion on the architecture.

6.3.2 Persistence to Goal

For the baseline architecture, average percent time spent near the reward platform for trained rats was 5.51. For the categorical limits architecture, the average was 5.48. As reported in results section 5.3, p-value for ANOVA of the baseline vs. limits architectures was 0.002254. Given the p-value, even though the difference in average is small, a significant effect between the architectures is concluded.

	N	E	S	W	Average Differences
N	0	169	-321	1206	264
E	59	0	64	64	47
S	-178	-141	0	-147	-116
W	352	1083	-163	0	318

Table 6.3: Differences of differences of pathlenths between naive release and releases with potential transfer learning of baseline architecture vs. categorical limits architecture.

6.3.3 Rapid Transfer Learning

Table 6.3 shows the difference between values reported in Table 6.1 and Table 6.2. Those last two tables showed the occurrence of transfer learning in the two architectures. Table 6.3 shows the increase in average transfer learning from the baseline architecture to the categorical limits architecture. In three out of four cases, there was an average improvement in transfer learning for the categorical limits architecture as compared to the baseline architecture.

Of the three Morris water maze tasks, rapid transfer learning is the task which requires the greatest degree of generalization ability in order to perform well. In the basic and persistence to goal tasks, the rat is attempting to perform a task for which it was trained. In the rapid transfer learning task, the rat is attempting to perform a task similar to a different task for which it was trained. E.g. the rat tries to navigate to the platform from the north release point, after having been trained form only the south release point. This task can be seen as a test of the generalization of navigation in a given tank. Navigating to the platform in general is an abstraction of navigating to the platform from a specific release point.

Given that limitsART is a modification to support generalization based on categorical limits, the rapid transfer learning task results for the basic architecture are compared to the rapid transfer learning task results for the categorical limits supporting architecture. Results section 5.3 shows the results from an analysis of variance on the non-limits versus the limits architectures. A p-value of 0.0124 is observed for

the ANOVA with null hypothesis that the non-limits and limits architectures will perform the same. Therefore, there is a statistically significant difference in behavior between the non-limits and limits architectures.

To help characterize that difference in behavior, the p-values for the individual non-limits and limits architecture based experiments are noted. The limits architecture performance is more significant, meaning that the null hypothesis is even less likely to have been falsified by chance. While the non-limits architecture is considered significant with a p-value of 0.04026, a p-value of 0.01240 for the limits architecture performance shows even lower chance that the measurement distribution might have occurred at chance.

In addition to the increase in statistical significance of the limits architecture, the beginning of this section shows that the average transfer learning performance of the categorical limits architecture is increased as compared to the baseline architecture. Results section 5.3 shows the p-value for the direct per-rat comparison of baseline versus limits architectures to be 0.003462. Therefore it is concluded that the categorical limits architecture results in improved transfer learning performance.

6.4 Comparison to Biological Experiments

A publication by Hamilton, Akers, Weisend, and Sutherland [107] provides data for biological rats performing the morris water maze task in general, and the persistence to goal morris water task in specific. The experiment in the Hamilton et al. study, which demonstrates persistence to goal behavior, began by training a sample set of rats on each of four possible release points around the perimeter of the tank. Release point order was selected randomly without replacement during each block of four trials, just as seen in the simulated training for the experiments reported in this dissertation. Rats were released near the wall of the pool and were retrieved from the platform five seconds after mounting it. If a rat did not find the platform within

sixty seconds, it was retrieved and placed on the platform for five seconds.

The experiment protocols exercised on the computational rats also retrieve rats which do not find the platform after a period of time. However, simulation of the placement of retrieved rats upon the platform is not performed. Doing so might improve learning convergence time. However, the published protocol does not specify whether retrieved rats are picked up and navigated to the platform oriented as if they were swimming or if they are picked up, carried through an arbitrary path, then deposited back on the platform. Given some uncertainty on how to perform the retrieval and replacement upon the platform, retrieval is simply performed without placement. Experimenting with platform placement after retrieval will be suggested in the future work section.

After completion of the final training trial, a thirty second probe trial was conducted with the platform removed. From the sample set of trained rate, half of the animals for each combination of pool position and platform location were assigned to either a Shift condition, where the pool was moved 75 cm (from the trained position to the other pool position), or to a No Shift condition, in which the pool remained in the trained position. This resulted in sixteen rats being tested in the Shift condition and sixteen rats being tested in the No Shift condition.

Hamilton reports a statistically significant effect for decrease in latency to navigate to the platform across trial blocks. As there is no meaningful way to label simulated rat speed in the simulated embodiment, constant speed for our embodied rat is assumed and therefore latency from release to platform is proportional to pathlength from release to platform. Pathlength is in fact what is measured. So, given an equivalent experimental protocol between simulated and biological systems, significant findings reported in section 5.1.1 and 5.2.1 correspond to behavior in biological morris water maze experiments.

6.4.1 Persistence to Goal

For his no platform probe trial, Hamilton finds that all 16 rats in the No Shift condition navigated to the absolute spatial location of the platform during training. Specifically, he finds a significant effect that rats spent more time in the region where they were trained to discover the platform, when the platform was no longer present. This is the same effect seen in section 5.1.2 and 5.2.2.

6.4.2 Rapid Transfer Learning

One of Morris's [108] experiments demonstrated what he labeled "instantaneous transfer." Rats were repeatedly trained from a single fixed release location in the Morris water maze task, and then exhibited better than untrained performance when released from a different location. Results sections 5.1.3 and 5.2.3 display statistically significant performance of the embodied non-limits and limits neural architectures, respectively, on this same task. As discussed previously, the non-limits architecture shows marginally significant performance. The architecture which supports categorical limits shows clear improvement over the non-limits architecture, and clear statistically significant behavior on the simulated transfer task.

6.5 Comparison to Other Models

In this section results are compared to the other distal reward models introduced in 2.1.2. Results are compared to highlight unique contributions of this dissertation research in the context of similar research. It is shown that many of the comparable models differ in their degree of embodiment from the dissertation model. These models are based on a variety of theoretical underpinnings yet are often similar to each other in that they leave incomplete on of the fundamental concepts that this dissertation is built upon; this is embodiment.

6.5.1 Izhikevich

Izhikevich models cortex at a high resolution, where the level of detail is a single neuron. Izhikevich's model is of the spiking neuron type. In this respect his model is two levels of specificity deeper than ours. The first level of abstraction is from action potential spike frequency to neural node activation. The second level of abstraction is from individual neurons to functional sub-assemblies such as ART and temporal integration arrays.

Despite the differences of the two models, they are both computational neural models which display distal reward behavior. Izhikevich demonstrates distal reward performance at the level of individual neurons by showing that a greater fraction of neurons from within a network will fire in response to a particular stimulus. That particular stimulus received a delayed reward in training, hence learning to respond to that stimulus demonstrates distal reward learning. In contrast, this dissertation research demonstrates distal reward performance at the embodied behavior level. While distal reward learning at the level of sets or sub-assemblies of neurons is necessary for distal reward learning behavior, unless embodied experiments are performed there is no guarantee that a particular form of neural distal reward learning will in fact result in embodied distal reward behavior. For instance, in the Izhikevich model, learning is evidenced through an increased proportion of neurons firing. However, in order to activate a given behavior there might be one particular neuron which needs to fire. Just because an increased proportion of neurons fires does not mean that a particular neuron fires. Stated another way, the Izhikevich experiments demonstrate that a particular scheme of neural learning can exhibit distal reward learning. However, it has not been demonstrated that there is any possible way of connecting the demonstrated neural circuitry such that an agent with this type of neural circuitry would be able to exhibit distal reward learning behavior. The contrast with this dissertation research is that we consider embodied behavior to be the goal.

6.5.2 Samsonovich and Ascoli

A commonality of this dissertation research with the Samsonovich and Ascoli work is that they both exhibit spatial demonstration of distal reward behavior. A constrast is that the Samsonovich and Ascoli experiments do not embody the model in the sense of having a neural sensory apparatus, though the models do have some embodied qualities. The concepts of location and movement, which are present in the Samsonovich and Ascoli model, are an important component of embodiment; however there is no sensory input to the model. The model is informed of location through direct stimulation of CA3 place cells. This direct location signal can be compared to a sort of GPS (Global Positioning System) feed in that no neural processing is required to determine location. A level of disconnect, with neuro-physical motivation, is established when this sort of direct neural stimulation is used. This need not be considered a shortcoming of the Samsonovich and Ascoli work because they do not seek to explore the representation of observed environments in neural architectures. The contrast with this dissertation research is that such exploration is a component of the embodied goal. One of the roles of embodied neural networks is internal representation of the observed environment. One reason this internal representation is important is because all learning is performed in the context of that internal representation. An embodied neural network does not know that a bound memory is germane to the current situation except that the current internal representation of the observed environment is similar to a component of the memory. Because the environmental representation is an integral part of embodied neural operation, this dissertation research uses visual sensory activation by simulation of light reflected in the environment.

Apart from the environmental representation difference, there is a similarity in model operation, as well as a couple more differences of note. A similarity is that both the embodied dissertation architecture and the Samsonovich and Ascoli model start with random exploration which leads to reward discovery and path learning to

that reward. A difference in the path learning is that the Samsonovich and Ascoli model binds memories at various arbitrary times during exploration, where as we bind a new memory whenever an existing memory component is experienced. The advantage of binding memories only upon experience of an existing memory is that each new memory has a context in which it exists relative to an existing memory. I.e. each new memory extends a chain of memories which encode navigation to the platform. In contrast, the Samsonovich and Ascoli method results in tangle of memories which encode overlapping path segments, or may encode disjoint path segments. Use of this tangle of memories will require further processing. Use of the dissertation memories which each link up to another memory is a more simple matter of letting each memory cue the subsequent memory.

A final enumerated difference is that the Samsonovich and Ascoli model chooses movements by selection of the best direction from 10 random choices. In contrast, the dissertation model chooses movement by selecting the best direction from all possible directions. Selection among random possibilities would potentially result is situations where a vastly better navigation choice is overlooked only because it did not appear in the random options list.

6.5.3 McKinstry et al.

McKinstry et al. [1] do not explicitly model hippocampal form or distal reward function. The Darwin IX model demonstrates temporal delay based ability or behavior through two model strategies. Temporal delay in neural circuitry is necessary for temporal reward behavior because the temporal delay allows sensory observation signals to persist in the neural system long enough to affect non immediate behavior. In this sense, the Darwin IX models addressed similar concerns as this dissertation, though McKinstry et al. did not extend their experiments to distal reward behavior.

The other comparison between this dissertation research and the McKinstry et al. research is the embodiment method. The Darwin IX embodiment is without

doubt complete because it is accomplished through a physical robot. Compared to the dissertation embodiment, the physical embodiment has some qualities that are more useful in generating experimental evidence for linking modeled neural structure to behavior, while other qualities make the physical embodiment less useful. Physical embodiment results in environmental response to agent action that is as exact as possible, i.e. it is based on natural physical consequence. On the less useful side, non-ideal mechanical factors are present in the feedback chain from action to environmental change to modified sensory input. Unintended lag caused by lose gears might confound intentional lag programmed into the neural model. Exact reproduction of experiments is also adversely affected by physical reality. A pebble may be rolled over in a certain location in one trial with the pebble being deflected by the robot wheel into another position that affects another trial in a different fashion. Another possible experimental result confounding factor in robotic embodiment is the affect of battery voltage on motor response. The embodiment method of this dissertation leads to experiments which are exactly reproducible any number of times without the slightest behavioral deviation that is not due to differences programmed by the experimenter.

6.5.4 Dolle et al.

A strategy used in the Dolle et al. [26] model, and avoided in the dissertation model is short-cutting elements of sensory perception. For instance, the Dolle et al. model provides grid cell activations directly within the hippocampus as opposed to deriving grid cell activation from sensory input. The Dolle et al. model uses sensory activations internally as well, but the direct grid cell activations are concatenated to the sensed visual activations. As such, experimental evidence linking neural model form to embodied behavior function is weaker in the context of biological neural systems. Biological neural systems receive no direct neural activation from external analysis of the environment, all neural activation is derived at some point from sensory activa-

tion levels. Full simulated embodiment is one of the fundamental motivations of this dissertation research, and is a differentiator for experiments such as from Dolle et al. which provide some or all input as direct neural manipulation. A further differentiator between this dissertation research and the Dolle et al. experiment is that the Dolle et al. model uses a global planning method to determine a full path between current state and goal state, where as the dissertation model performs incremental decision making as sensory observations occur along segments of the path to reward.

6.5.5 Weber and Triesch

A main contrast between the Weber and Triesch [33] results and the dissertation results is that the Weber and Triesch results can not be said to reflect upon a link between neural form and embodied neural system function. Barriers to such a claim are the small torus environment and the non-deterministic environment response to agent action. Differentiators were pointed out between some of the previous models and this dissertation model which involved the model itself. While the are also differentiators between the dissertation model and the Weber and Triesch model, the differentiator we point out now is the simulated experimental environment. By simulating a realistic if simplified environment, this dissertation makes possible the application of experimental results to the link between neural form and embodied function, as well as comparison to distal reward learning as performed by biologically embodied neural systems such as rats.

6.5.6 Martinet et al.

The Martinet et al. [27] model is another example of incomplete embodiment through direct neural stimulation based on external environment analysis. Namely that external environment analysis is the identification of the embodied agent at a certain location. Unless equipped with a neurally interfaced GPS (Global Positioning Sys-

tem), biological neural systems lack this type of direct knowledge. Considered another way, full embodiment would require that sensory observations originate from the same environment in which actions are performed. With a simulated robotic platform and direct place cell activation, action occurs in the simulated maze, but there is no sensory observation. If there were said to be sensory observation, it would be in an environment of discrete place cells, instead of the environment of maze walls and floor where action takes place. The Martinet et al. experiments are another published example of a high quality model that falls slightly short of full embodiment. As discussed earlier, one motivation of this dissertation was to close that gap in computational experimental practice.

This chapter has discussed in detail the experimental results of this dissertation. The next chapter will summarize this program of research and make concluding observations.

Chapter 7

Summary and Conclusion

The high level contribution of this dissertation is to increase the understanding of neural mappings between form and function. Form was investigated through category theory and study of neuroscientific literature pertaining to biological structures. Function was investigated through study of episodic memory and development of a temporal semantic representation theory. Understanding of the form function link was increased through computational experiments which allowed the test that certain neural forms would perform certain functions. Distal reward learning is the function which was selected for investigation. Episodic memory is a class of memory which is strongly implicated in distal reward learning. Hippocampus is an element of neural form which is linked through the biological literature with episodic memory and therefore distal reward learning. An artificial neural architecture was engineered and embodied in simulation. This embodied architecture is the engineered form which this dissertation linked back to function through experimental evidence. Therefore, understanding of the biological neural form function link between hippocampus and distal reward learning was increased through demonstration of a constructed form function link between computational hippocampal form and simulated embodied distal reward function. Through this process, evidence was generated to demonstrate a nascent mathematical theory that links mathematical form, i.e. structure, with

neural function. The mathematical form introduced was categorical limits, which describe generalization, and the neural function was generalization in the distal reward task. Again, the theoretical form function understanding was strengthened through embodied experimentation in the same sense as the theoretical understanding of aerodynamics is strengthened through an operational airplane. In addition to the described form function links, this research theorizes, develops, and demonstrates a method for motor control based upon cyclic bottom-up top-down neural function. Furthermore, this research theorizes, develops, and demonstrates a navigational technique using episodic memories of expected reward. The combination of the motor control and navigational methods enabled a functional embodied agent. The following paragraphs describe conclusions from this research which comprise details from the high level contribution.

It was posited that a biophysically motivated neural architecture which models episodic memory enabling aspects of the cortical-hippocampal system from mammalian brains will have the capacity, when embodied in an agent, to exhibit distal reward behavior. Furthermore, the Categorical Neural Semantic Theory was to be applied and directly suggest architectural changes to the system that increase performance. In the preceding text, neurophysical motivation was presented and a stand-alone cortical-hippocampal model was implemented. The stand alone cortical-hippocampal model was experimented upon to provide evidence of behavioral correspondence with its biological basis. That cortical-hippocampal model was refined and incorporated into a computationally embodied artificial neural architecture. This embodied architecture performed a well researched distal reward task which is extensively documented in neuroscience literature. Numerical analysis demonstrated in a statistically significant sense that the artificial neural architecture exhibited behaviors which are observed in the corresponding biological rat experiments. Category theory was introduced and experimental investigations presented. An augmentation was implemented which incorporated category theory derived theoretical advance-

ments to the ART modules used in the embodied neural architecture. Statistical analysis showed improved performance for the category theory motivated architecture change. This improved performance was in a generalization task, which is exactly the theoretical capability which the categorical improvement added to the artificial neural architecture.

Through the example of this research, it can be concluded that category theory has performance enhancing design input in regard to embodied neural structure. As well, it has explanatory power for the analysis of neural structure in general. The use of category theory was demonstrated through the initial categorical analysis of standalone neural networks, then the demonstration of performance enhancement through the addition of categorical limits to a computationally embodied neural architecture.

It can also be concluded that modeling episodic memory encoding structures from biological brain is sufficient for embodied performance of a distal reward task. This piece of knowledge contributes to computational synthetic neuroscience as well as to the body of engineering theory which researchers interested in implementing related systems can draw upon.

As a novel navigation scheme, binding and recall of proprioceptive temporal episodes with environmental sensory temporal episodes has been demonstrated to successfully form the basis of embodied locomotion. Stemming from experimental validation of the theoretical navigation method, it can be concluded that the proposal of a unified sensory motor system incorporating both top-down and bottom-up neural function was likewise validated. This scheme can be extended to more advanced behaviors, and obviates the need for complex neural reverse kinematic processing, with applications in both virtual and physical robotics, locomotion, and navigation. A prediction follows from this motor control proprioceptive recall scheme. Biological organisms which are denied sight during development never develop certain visual brain structures such as striate cortex. It is predicted that organisms which are denied movement during development will never develop certain brain structures in

motor and premotor cortex. Furthermore, as a test of specific proprioceptive recall, it is predicted that organisms which are denied movement of select joints during development would never develop specific sections of motor cortex structure and the accompanying capability.

Finally, it can be concluded that computational neuroscience has contributions to make in investigation of the form function problem. This research is one spot illumination upon the neural form function problem and is a scientific contribution toward the theoretical basis that it is hoped will one day result in realization of true general computational cognition.

Chapter 8

Future Work

This research has the potential to generate a great deal of future work. We hope to expand this research along several avenues. Structure based understanding of brain function is an especially promising topic. Higher fidelity models of hippocampus and supporting cortex can be constructed through iteration of the current design. New brain areas can be modeled and tested. Predictions have been produced that can be followed up with new experimentation and analysis of data from biological organisms.

8.1 Structure

Structure based understanding of brain function could refer to any types of structure. Specifically, we assert that there are many relationships to be discovered between categorical structure, brain function, and behavior.

A point is made of calling out categorical structure because there is much mention in the literature of the "form function link." Though form and structure are synonymous, almost no biologically focused literature to date is concerned with mathematical structure as treated by category theory. In the term form function link, form is referring to actual neural structure as opposed to the abstract categorical structure

we wish to investigate. Abstract structure must be related back to neural structure, an effort which this research initiates in the embodied domain. It is suggested that abstract categorical structure is a rewarding level at which to investigate the link with neural function, and further research should be pursued.

8.2 Feedback in Reward Realization Motor Plan

An opportunity exists to introduce sensory memory feedback into motor recall processing. In operation, each motor plan that is bound to expected reward leads to another such that successful navigation occurs. However, within each plan the method is to initialize motor recall and follow the bound motor memory. An improvement to following the motor plan would be to initialize sensory recall in parallel with the motor recall.

Just as motor sequences are encoded by the ART - temporal integrator - ART somatosensory cortex neural assemblies, sensory sequences are encoded by by the ART - temporal integrator - ART sensory cortex neural assemblies. Trajectories through motor space which are encoded as the proprioceptive sequence component of episodic memories (i.e. motor paths) should correspond to trajectories through sensory space which are encoded as the visual sequence component of episodic memories. Error correction techniques could be employed to drive the actual sensory trajectory to the recalled expected sensory trajectory. Memories only encode characterizations of experienced episodes so feedback would not results in exact error correction but it is suggested that performance would be improved.

We propose an experiment which would provide evidence as to whether biological rats use environmental sensory feedback during navigation to learned reward. The experiment uses two environments which are indistinguishable until a certain location threshold is crossed. After that threshold, some form of sensory input would differ. For instance, solid floor tiles could be used throughout the first environment,

solid floor tiles used for the first half of the second environment, and hollow floor tiles used for the second half of the second environment. A given factor is that rats can sense the difference between solid and hollow floor tiles. The protocol is for a rat to learn to navigate to reward in one environment, then undergo testing in the second environment. Because the second environment is indistinguishable from the first for the initial portion of navigation to reward, the trained rat would begin along the learned path. Some portion of the way through that path, environmental observations would cease to match the environmental observations which were present when the sensory-motor episodes were formed. If the rat simply continued along the learned path, there would be evidence that motor recall is being initiated then followed regardless of environmental matches to other components of the sensory episode. It is noted that the dissertation model uses many episodic memories in navigation from start point to reward platform, so there are in practice sensory checkpoints all along the total navigation path. Still, the dissertation model would not alter behavior if sensory input changed mid-navigation segment. In the proposed biological experiment, if the rat changed behavior the instant the new sensed environmental condition occurred there would be evidence that multiple components of memory episodes are recalled and compared to current input.

8.3 Morris Water Maze Rat Platform Placement After Retrieval

As discussed previously, when the simulated rat has been searching for the hidden escape platform in the water for too long, we retrieve it as is customary with biological rats. Some biological experiments proceed to place the retrieved rat on the escape platform for a period of time after retrieval. For the purpose of future work, we hypothesize that platform placement after retrieval would decrease average convergence time for training. The specific protocol for placement would be to, upon

the time limit being reached and the rat not having found the platform, manually locomote the rat from its current position directly to the platform. This would be equivalent to biological experiments where the rat can sense its environment as it is moved to the platform. Through the process of externally administered locomotion, rat orientation would be as if the rat were swimming itself directly to the platform.

Manual locomotion after retrieval inspires a further set of experiments. Figure 3.6 illustrated the expected reward episodic memory bindings which the embodied architecture forms. An ideal expected reward visualization would have all arrows oriented directly at the platform, indicating that the highest expected reward condition from any location in the tank is being set up to move directly toward the platform. A measure of the potential for any particular embodied architecture to encode that ideal could be taken through manual locomotion. If the rat were repeatedly trained from all points and orientations in the tank, how to navigate directly to the platform, an architecture with unlimited representation ability and capacity should form the ideal expected reward map. Ability and capacity of subsequent architectures realized with some constraint on resources could be expressed as distance from ideal.

8.4 Decouple Environment Learning from Distal Reward Learning

One way in which the experiments reported in this dissertation were made more difficult than necessary was that the computational rats began training in an almost completely naive state. The computational rats did recognize the platform on contact, but they did not have an perceptual primitives with which to understand the environment. Perceptual primitives could be considered at many levels. For a person, high level perceptual primitives with which to form navigation episodic memories might include doorways and street or building signs. Temporal episodes of observations can be built from previously learned patterns which are observed. For

example, when I build a memory to navigate down the hall, I need not store each doorway as an arrangement of colors and shapes in my vision, I already know what a doorway is. The computational rats do not already know what anything in their environment is, their modeled visual cortices are blank.

An experimental extension would be to pre-expose the rats to their simulated environment. The rats could simply wander for a time and build up sensory input pattern based templates. Allowing the rats to begin training with some experience of the sensory world in which they will exists might allow for more nuanced experiments regarding distal reward learning as differentiated from sensory learning.

8.5 Hippocampus

Another future goal of this research is to incorporate a higher fidelity version of the hippocampal model into the embodied architecture. In the current architecture, we faithfully model hippocampus as a location of cortical convergence, and a closed loop where memory merges with recall. Some functional elements of hippocampus though were refined out in creation of the embodied architecture. Extended research could make use of the full hippocampal complexity that was developed in the model which incorporates endorhinal cortex, dentate gyrus, CA3, and CA1 with neuromorphic connectivity. The model fidelity could be further increased by separating out frontal cortex structure and function. Frontal cortex is similar to hippocampus in that is has afferent and efferent connections with a convergence of cortical regions. Frontal cortex is not implicated nearly as strongly in episodic memory as is hippocampus, but frontal cortex would likely play a role is decision making as guided by bound episodic memories.

8.6 Further Brain Modeling

In addition to more detailed models of hippocampus, there are other brain areas that can be modeled in detail. Figure 8.1 illustrates an expanded diagram of brain areas involved in episodic memory behavior. It would be interesting to model some or all of these additional areas and functionally tie them in to embodied behavior. For instance an Amygdala component should enable an emotional facet to embodied behavior. An emotional response would open the potential for some memories to form more strongly than others due to environmental cues causing a response such as fright. Neural form that supports emotional function is an important factor in research into disorders such as post traumatic stress disorder.

8.7 Developmental Perception

Another offshoot of this research that we would like to pursue further is developmental perception. Building upon the categorical conceptual capacity research in section A.4 along with the neuromorphic principle of overlapping input fields, we have shown the development of perceptual primitives. Overlapping input fields within the same sensory modality is a principle that was partially refined out of the final embodied architecture, though overlapping input fields between distal and proximal modalities was preserved. This offshoot of research is satisfying because it allows the exploration of overlapping input field concepts further. An initial experiment is presented to show the promise of this avenue of future inquiry.

For this experiment, our architecture has overlapping input fields as in Figure 8.3. There are two sources of motivation for overlapping input fields, biological and categorical. We find biological motivation in the connective structure of simple cortical cells. As illustrated in Figure 8.2, simple cortical cells have rectangular receptive fields that respond to visual line segments in different orientations. Each simple cortical cell represents a different orientation. Multiple simple cortical cells,

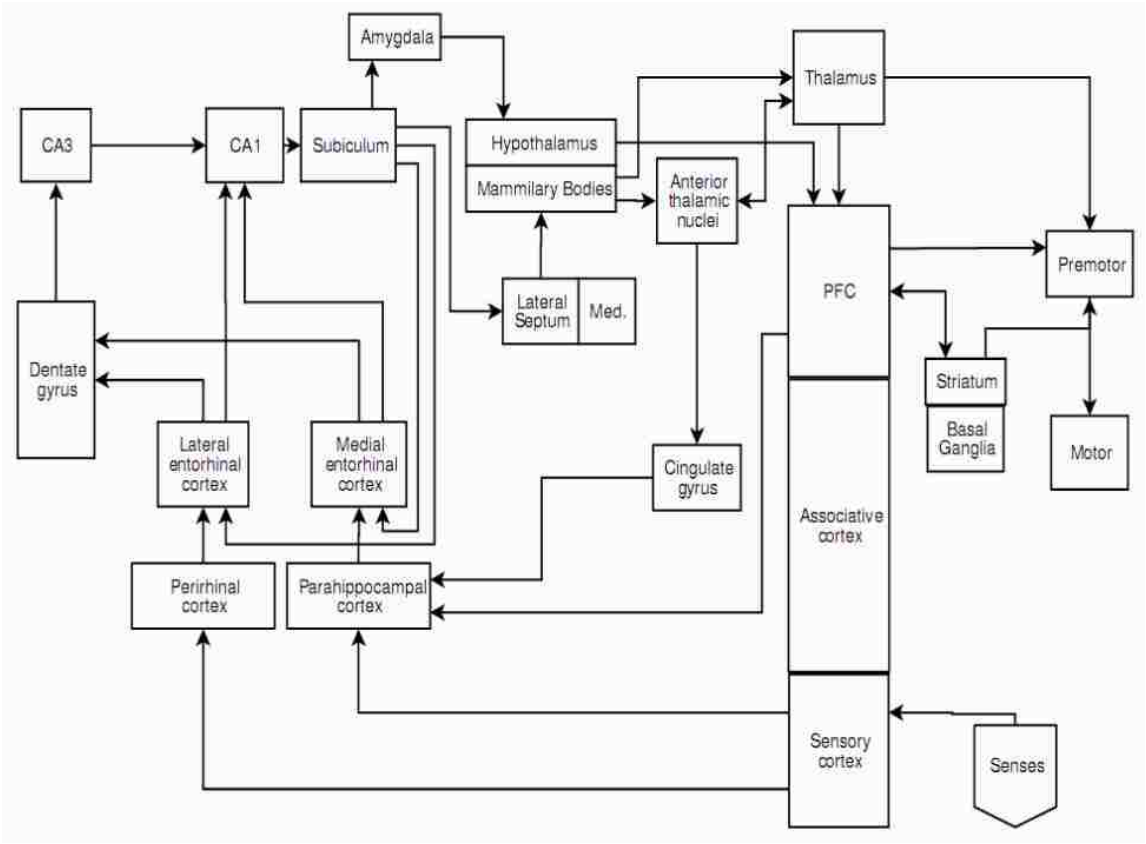

Figure 8.1: Extended functional brain diagram from which embedded architecture is distilled.

representing different line orientations, receive stimulus from some of the same retinal ganglion cells[18]. Hence, the receptive fields of the simple cortical cells overlap. An additional function of separate neural functional subsections that take overlapping input is the implementation of parallel search for specific perceptual primitives (short line segments at specific orientations). The parallel speed-up is in comparison to one large classification system that attempts to match a set of line segments across the whole visual field at once. A general one-dimensional overlap is implemented so as not to bias the form of perceptual primitives that the synthetic organism will find. The form of overlap seen in the biological system is biased for oriented line segment identification by a simple neural perceptron. Reasonably, this form of architectural

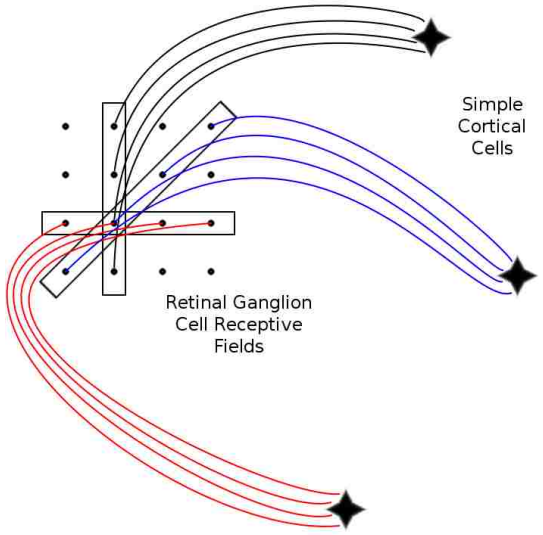

Figure 8.2: Illustration of connection architecture between retinal ganglion cells and V1 simple cortical cells.

bias toward a certain perceptual primitive was probably developed through fitness and selection over evolutionary time.

For this investigation, we implement an architecture of multiple, relatively small, retinotopically mapped, neural modules with overlapping input fields. We use ART to classify the inputs of each module into an output vector. The experimental architecture of small overlapping ART units has 32 ART units covering a 99 element array of visual sensors. The 99 visual sensor elements are uniformly distributed across the embodied synthetic organisms visual arc. Each ART unit inputs 6 visual sensor elements. Each visual sensor element has a red, green, and blue component. The inputs to each ART unit are complement coded. Therefore, each ART unit has an input vector of 36 real values. Each ART unit overlaps by half with its neighbors, with the end units only overlapping on one side. Figure 8.3 illustrates the architecture scheme, though not the specific number of ART units and inputs used for this experiment.

To analyze formation of sensory perceptual concepts learned from the observed

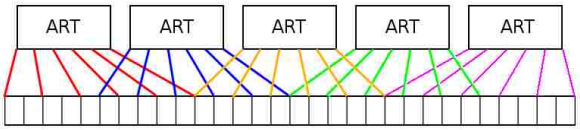

Figure 8.3: Simple one layer architecture, overlapping input fields.

world in the ART units, we non-parametrically cluster all formed templates. Nearest neighbor clustering is used on the vector of connection weights that encodes each template. Clusters represent perceptual concepts formed in the ART units. Analysis of which ART units have a member template in a given cluster provides information as to whether the cluster concept is a universal concept in the context of the simulated world, or if it is a somewhat unusual concept only seen in certain conditions and not likely to be useful as a perceptual primitive.

The more ART units in which a particular template is found, the more likely that template is to represent a perceptual primitive. For instance, if a line segment is a perceptual primitive for a particular system, then we would expect to see neural patterns that represent line segments occurring in locations throughout retinotopically mapped visual cortex. In biology, the simple cortical cell encodes line segments and is found encoding line segments is all locations throughout the visual field[18].

In section 8.7, the percentage of ART modules in which each template cluster is found is analyzed. In biological visual cortex, it would be expected to find the equivalent of clusters that represent oriented lines in 100% of visual field subsections. If our architecture displays common templates across its visual field, then a reasonable claim can be made to an analouge of developmental perception. The developmental claim is valid in part because the ART modules are self-organizing, they start with no designed in notion of a common perceptual primitive. Therefore, any common perceptual primitives which from across sections of the visual field must be a result of the simulated developmental process.

The developmental perception analysis looks for evidence of perceptual primitives formed in our artificial neural architecture. Each template, in an ART that encodes

a portion of the visual field, represents a certain pattern of inputs in that visual sub-field. Templates are encoded as weight vectors in the ART modules. As such, it is possible to take a measure of euclidean distance between two templates. With a distance measure, it is possible to cluster templates, and that is how we analyze the results.

100 96 96 100 100 100 93 100 12 25 84 100 18 3 9 100 18 12 18 3 3 18 15 21 3 6 12 6
9 3 6 21 56 3 9 3 9 9 21 6 6 15 9 6 3 3 3 3 3 3 3 3 3 6 9 3 6 3 3 3 3 3 9 3 6 25 9 3 6 6 9
3 3 3 0 0

Figure 8.4: Percent of templates represented per cluster, 30 degree visual arc.

100 96 100 100 100 100 100 100 100 100 100 100 96 9 18 12 12 3 21 15 12 3 21 3 12
25 6 3 3 3 3 34 18 25 21 12 9 15 3 3 3 3 3 3 3 3 3 3 3 3 3 18 6 15 3 12 3 21 6 15 3 3 6 3
6 18 6 28 6 3 6 9 6 3 12 9 3 3 3 3 3 6 3 3 3 3 3 0 0

Figure 8.5: Percent of templates represented per cluster, 1.25 degree visual arc.

Figures 8.4 and 8.5 show the percent of all ART units that have at least one member template in a given cluster for two different visual arc angles. We see a number of clusters with members in all ART units. These clusters give evidence that universal perceptual primitives are represented. With evidence that perceptual primitives can be represented at the lowest level of the architecture, more complex architectures can be considered.

As a corollary to the investigation on developmental perception, experiments are performed on a variation of ART that implements forgetting. An additional constraint is applied to the environment where ART is used in an embodied system. Our system is resource limited and therefor has a maximum number of F2 nodes with which to form abstractions. The main consequence of this constraint is realized when highly infrequent, and unusual, inputs are considered. Highly infrequent means that the inputs may be encountered only a few times in a simulated lifetime. Unusual

means that the inputs are not similar to any other inputs encountered. The ART orienting subsystem will make it likely that any unusual input gets a new abstraction. If that input, or anything like it, is never observed again, then the F2 node used to form it's abstraction is wasted in the context of the rest of the system's life and a fit system with constrained resources would forget.

8.8 High Performance Computing

Yet another of the research avenues that we would most like to investigate is the implementation of a high performance computing (HPC) version of the architecture. The current architecture was implemented using OpenMP to parallelize ART's resonant search for a winning template. However, on a modern four core processor, the overhead of data distribution to parallelize the search only breaks even with the realized speed-up, giving no net gain.

Similarly, the 128-bit registers available in modern processors were used to perform four single precision float comparisons at one. Using this parallelization during the ART template search has the potential to give the same speed-up as OpenMP on a four core processor. Unfortunately, the overhead of packing and loading four floats into the 128-bit register also canceled out the gain from the parallel search.

These negative results suggest that greater than four way parallelization is required to successfully amortize overhead costs of distributing computational tasks. We would like to investigate MPI on HPC clusters as a traditional avenue of massive parallelization. Also, we would like to investigate general purpose graphical processing unit processing as an efficient means of large scale parallelization.

8.9 Robotics

It is proposed that the demonstrated system of proprioceptive sensing and recall would be well suited to physically implemented systems. Especially for mechanically complex systems, reverse kinematics is a difficult problem. The experience based system in this dissertation could greatly simplify novel motor path realizations. In essence this simplification is because no paths would actually be novel, all movement would be built up from previously experienced proprioceptive states. The expected reward mediated system used by the embodied architecture could also be employed to build complex motor paths from simpler experiences. For instance, pairwise training among degrees of freedom might be chainable in recall. For instance in a robotic arm with three rotational joints like a wrist and three angular joints like an elbow there are six degrees of freedom. A relatively complex motor path would involve moving all six joints at once. A relatively simple motor path would involve movement of two joints. There could probably be an algorithm developed where joints that move in sequence along a motor recall path would learn to move in parallel for efficient arm positioning.

This ends the Future work chapter. A number of avenues for extensions of this dissertation research were suggested. The remainder of this document is the Appendix section and References list.

Appendix A

Introduction to Category Theory

Category theory is the mathematical discipline concerned with structure. The preceding explanation demonstrates our rationale for considering structure to be a critical consideration in the design or modeling of any neural system. Category theory provides a way to treat structure with mathematical rigor.

The following subsection provides a brief introduction to some of the basic ideas of category theory. A more in depth introduction is provided in appendix A. The introduction in the appendix explains in more detail the foundational ideas of category theory. Though from the point of view of a practicing category theory mathematician, even the appendix introduction barely scratches the surface. The reader who is interested in an even deeper journey through beginning category theory is referenced to [62], [109], and [110].

A.1 Category Theory from Thirty Thousand Feet

Some of the primary ideas of category theory are *objects*, *morphisms*, *categories*, *functors*, *natural transformations*, and *commutative diagrams*. *Objects* are the atomic unit of category theory, they can represent anything, physical or nonphysical, real or imagined. *Morphisms* are transformations between *objects*. *Morphisms* detail the

relative structure of *objects*. *Categories* are collections of *objects* and *morphisms*, one might have a *category* of vegetables or a *category* of theories. *Functors* are transformations between *categories*, they map each *object* and *morphism* in a domain category to *objects* and *morphisms* in a codomain *category*. *Natural transformations* are transformations between *functors*. If a *functor AB* maps *category A* to *category B* and an other *functor AC* maps *category A* to *category C*, then *natural transformation* AB_AC can be defined between *functor AB* and *functor AC*. *Commutative diagrams* are key structural elements in the use of category theory. A *commutative diagram* is made up of two equivalent, *morphism* defined, paths between a domain *object* and a codomain *object*.

A defining characteristic of functors and natural transformations is that they are structure preserving. A commutative diagram in one category will, when mapped to another category by a functor, still be a commutative diagram. The specific objects and morphisms involved will likely change, but the structure remains. This preservation of structure is why category theory has descriptive power for the structural representation of the world in a neural network.

As noted earlier, a deeper introduction to category theory is found in Appendix A. Also in Appendix A and Appendix E are the results of experiments using category theory to analyze neural networks.

A.2 A Comment on Structure

Any intelligent system must have some sort of representation of its external and internal environment. Even a simple sense-response system has an implicit world representation that tells it a certain response is appropriate for a certain sensation. An amoeba with a chemical food sensor and locomotive cilia forms a world representation that includes concepts of spatial geometry, "if food sensed in direction x, then movement along direction x will decrease distance to the food". In some sense,

a system is considered more intelligent the better job it does at correctly forming and manipulating a world representation. For example , a system that could form and manipulate a world representation where all observable phenomena such as time dilation and gravitational waves are explain would be considered quite intelligent. Category theory provides a mathematical language that allows discussion of the structures that must be formed and transformations that must hold true in a valid world representation.

The knowledge stored in our biological neural networks, i.e. brains, derives its meaning completely from structure in its sensory apparatus. Each neuron does not have a "label" that identifies its meaning. The meaning of a given neuron is defined by that neuron's connections to other neurons and their individual activations, that is, the position in the neural circuit and the posible range of inputs it may receive. The relationship between structure and function (i.e. meaning within the brain) is of great importance in neuroscience [111] and is part of what we explore in this work.

As an extended example: A dog walks across our field of view. A million sensory inputs per eye send signals into our brain [18]. Various portions of our brain extract information from those inputs. A given alignment of active inputs causes a simple cortical cell to activate [112] [113]. The activation of that simple cortical cell represents a certain edge in the visual field. Activation of the correct simple cortical cells causes activation of a certain complex cortical cell [18]. We reasonably hypothesize, without giving further exact details, that visual processing continues through many neurons and connections as higher level features are extracted. Four legs may be recognized within the image. The temporal pattern of inputs could be processed by a collection of neurons to detect a characteristic dog-like motion. Ultimately, a specific arrangement of neural activations forms in higher level collections of neurons that represents a dog. This is the result of many self organizing operations using no external labels on any arrangement of activations. The particular arrangement of neural activations only represents a dog because of the hierarchical structural rela-

tionships of all of the involved nodes leading from the sensory system.. The primary lesson here is that the semantics of any neural representation is dependent on entire hierarchical subnetwork, including sensor neurons, that leads to its activation.

All nodes and connections in a network, taken together, describe the structure of that network. As connections form the context by which neural node activations have meaning, all meaning in a neural network is dependent on the network's connections. Therefore all meaning in a neural network that can be represented is dependent on the structure of that network. As category theory is the study of mathematical structure [114], it is natural to use it to study meaning in a neural network.

A.3 Category Theory from Five Thousand Feet

Having provided a broad introduction to category theory in the last section, this section provides a more detailed description in the context of neural representation. Objects, morphisms, commutative diagrams, functors, natural transformations, and limits will be discussed.

A.3.1 Objects

The notation is used that a neural object (p_i, η), from the neural category, is a carrier node p_i in an output interval η over transition Φ. A node p_i is said to be active, relative to the object (p_i, η) if its output falls in η. The region of activation for p_i is the subset of input space for which p_i is in η. The neural object (p_i, η) is highlighted if its node p_i is active in η. Φ represents the total neural processing of input e; $(v, \psi) = \Phi(\omega, \theta, e)$, where ψ and θ are arrays of node activation values for the whole network and v and w are weight arrays. (θ, e) is called an instance associated with the transition Φ. Transitions are considered that, because they occur after a feedforward network has stabilized, leave θ and ω constant.

A signal path is a set of neural category objects and the neural network connec-

tions between the carrier nodes for the objects. The signal path relates activity in the neural network to morphisms in the associated neural category. Each signal path is of the form $\gamma = [(p_1, \eta_1), c_1, (p_2, \eta_2), ..., c_{n-1}, (p_n, \eta_n)]$ where (p_1, η_1) is the source object and (p_n, η_n) is the target object. A signal path γ has an associated set of carrier node activations θ. The activations θ correspond to node outputs $\xi = \phi(\theta)$ which place each carrier node p in its active interval as determined by the signal path γ. $\phi()$ is the activation function of a node which takes as input the activation θ and determines the node output $\phi(\theta)$. By calling out each neural object in the path, γ prescribes an interval for each carrier node on the path. A path set Γ consists of one or more paths $\Gamma = \{\gamma_1, \gamma_2, ...\} = \{[(p_1, \eta_1), c_1, ..., c_{n-1}, (p_n, \eta_n)], [(p_1, \eta_1), c_a, ..., c_z, (p_n, \eta_n)]\}$. Note that the curly brackets enclose the path set, square brackets enclose each path in the set, and parentheses enclose each object in the path.

A.3.2 Morphisms

A morphism $m : (p_i, \eta_i) \rightarrow (p_j, \eta_j)$ consists of a domain (source) object (p_i, η_i), a codomain (target) object (p_j, η_j), a path set Γ between domain and codomain objects, and a set $U_{\Gamma,w}$ of all the instances (θ, e) which allow the paths $\gamma \in \Gamma$ to persist over Φ. By persist, we mean that each node output is in the specified interval and each connection is nonzero before and after Φ. We call the instances (θ, e) in the set $U_{\Gamma,w}$ also instances for the morphism m.

A.3.3 Commutative Diagrams

A composition of morphisms is a pasting together of morphisms $m' \circ m$. Instances of $m' \circ m$ is the set $U_{\Gamma,w} \cap U_{\Gamma',w}$ where Γ' is the signal path set for m', see [62] for more details. For Figure A.1 to represent a commutative diagram, the composition of morphisms that form the path from p_1 through p_2 to p_4 must equal the composition of morphisms that form the path from p_1 through p_3 to p_4. De-

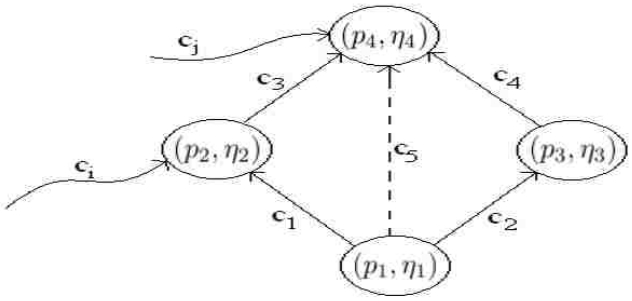

Figure A.1: Potential commutative diagram

fine morphisms $m_1 = \{[(p_1, \eta_1), c_1, (p_2, \eta_2)]\}$, $m_2 = \{[(p_1, \eta_1), c_2, (p_3, \eta_3)]\}$, $m_3 = \{[(p_2, \eta_2), c_3, (p_4, \eta_4)]\}$, and $m_4 = \{[(p_3, \eta_3), c_4, (p_4, \eta_4)]\}$. Then, the left side path $m_l = m_1 \circ m_3 = \{[(p_1, \eta_1), c_1, (p_2, \eta_2), c_3, (p_4, \eta_4)]\}$ and the right side path $m_r = m_2 \circ m_4 = \{[(p_1, \eta_1), c_2, (p_3, \eta_3), c_4, (p_4, \eta_4)]\}$. If $U_{\Gamma_l, w} = U_{\Gamma_r, w}$ then we have a commutative diagram because the right and left paths are for the same morphism. In that case, the intersection of the instances for m_1 and m_3 equals the intersection of the instances for m_2 and m_4, therefore $m_l = m_1 \circ m_3 = m_r = m_2 \circ m_4$. If there existed some c_i incident on p_2 then the morphism m_l could have a contributing factor lacking in m_r and the two morphisms could be different. If there existed some c_j incident on p_4 then we would be unable to guarantee the desired activation of p_4 based only on the connections from p_2 and p_3. These are not the only possible conditions which would prevent a commutative diagram, just representative examples. Figure A.1 can only represent a commutative diagram if instances of the objects and connections of m_r are one to one with those of m_l.

A.3.4 Functors

A functor is a transformation from one category to another that preserves compositions and hence preserves commutative diagrams. A neural network can be analyzed by using the neural category determined by the previous methods. One question

is whether the category associated with the neural network contains enough commutative diagrams so that a functor can be defined from **Concept** to the neural category. If a neural network can represent a commutative diagram then there exists the capacity to accept a commutative diagram mapped from the concept category to the neural category then induced onto the neural network. Using limits and colimit [62], we can apply commutative diagrams to represent the formation of both more specific and more abstract concepts from a given collection of concepts and concept morphisms. Therefore, we desire our neural networks to yield neural categories having the capacity to accept enough commutative diagrams to represent a nontrivial part of **Concept** in our neural network.

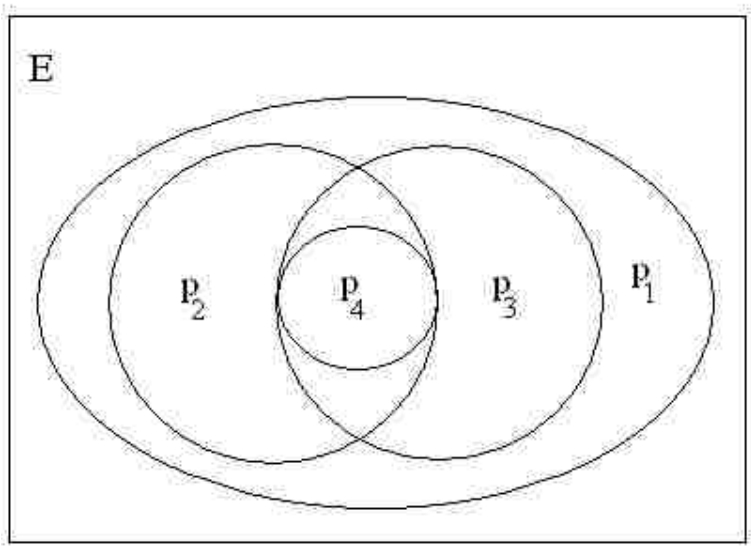

Figure A.2: Commutative diagram input regions. E is the set of all input space. The regions $p_1, ..., p_4$ are the activation regions, mapped to input space, for the objects $(p_1, \eta_1), ..., (p_4, \eta_4)$ from figure A.1.

Figure A.2 shows how the corresponding instance sets are related given that Figure A.1 is a commutative diagram. For m_r to be equivalent to m_l, p_2 can only

be active in η_2 when p_3 is active in η_3. If Figure A.1 corresponds to a commutative diagram that is the functional image of a diagram in **Concept**, there is an additional consideration: the instances of object (p_2, η_2) are also instances of the object (p_1, η_1) and similarly for the domain and codomain objects of the other neural morphisms. This consideration, along with the assumption that each neural object represents at most one object from **Concept**, leads to a nesting of sets corresponding to the instance sets of Figure A.2. This gives us the constraint that the region of activation for p_1 must contain the regions of activation for p_2 and p_3. (p_4, η_4) can only be active when (p_2, η_2) and (p_3, η_3) and active.

A.3.5 Natural Transformations

As functors are structure preserving mappings between categories, natural transformations are structure preserving mappings between functors. One application of natural transformations to biological neural networks is in multimodal systems. Each sensory modality builds up a conceptual structural representation of the world as observed through that modality. Given multiple structure preserving transformations, for multiple modalities, a transformation can be defined between those transformations. I.e. a transformation between the different sensory modality functors. A transformation among functors is a natural transformation.

A.3.6 Limits

Limits are abstract notions that can be related to specificity and generalization. For a collection of objects and morphism, a limit represents a unique property which all objects in the collection map from. The dual notion, a colimit represents a unique property which all objects in the collection map to.

An important use of these key notions is in the definition of limits and colimits. Healy and Caudell, [62] [67], show how colimits model the learning of more complex

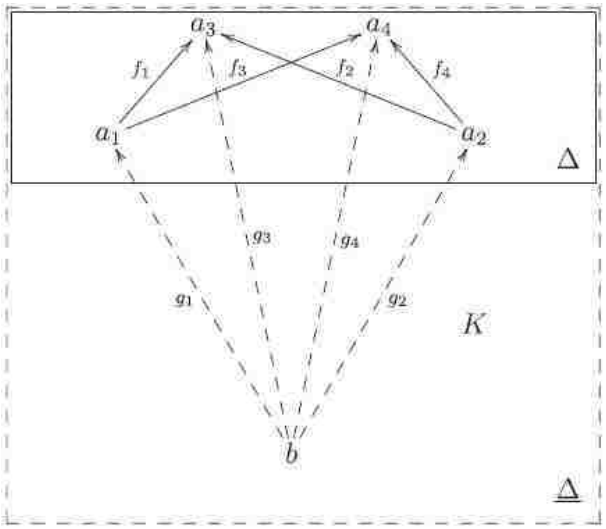

Figure A.3: A limit for a diagram Δ

concepts through the re-use of simpler concepts already represented in the synaptic-weight memory of a neural network. Their Axiomathes paper, [62], shows how limits model the learning of simpler, more abstract concepts through the re-use of existing representations. Reversing the arrows and substituting "initial" for "terminal" and "cocone" for "cone" in the following description of limits provide an overview of the dual notion, colimits, which corresponds to the learning of complex, more specific concepts through structures of more general concepts. Let Δ be a diagram in a category C as shown in Figure A.3, with objects a_1, a_2, a_3, a_4 and morphisms $f_1 : a_1 \to a_3$, $f_2 : a_2 \to a_3$, $f_3 : a_1 \to a_4$, $f_4 : a_2 \to a_4$. The diagram $\underline{\Delta}$ extends Δ to a commutative diagram by adding a cone K, consisting of an apical object b and morphisms $g_i : b \to a_i (i = 1, ..., 4)$ (called colimit leg morphisms) such that $f_1 \circ g_1 = g_3 = f_2 \circ g_2 and f_3 = g_1 = g_4 = f_4 \circ g_2$ (provided additional objects and morphisms with the requisite properties exist in C). Cones for Δ are the objects of a category $cone_\Delta$ (whose morphisms are described in Ontologies and Worlds in Category Theory: Implications for Neural Systems [62]). A limit for the diagram Δ is a terminal object K in $cone_\Delta$, in which case $\underline{\Delta}$ is called the defining diagram for the limit and Δ is called its base diagram.

A.4 Conceptual Capacity

In this section, the capacity of a given neural architecture to represent at least some portion of the concept category is explored. We concern ourselves with feed-forward neural networks that are fully connected by layer. Note that the selected architectures are not suggested as being of stand-alone practical use, but rather represent morphologies likely to be found in larger artificial neural networks. In order to explore the capacity of a given architecture, we conduct repeated trials with different weight vectors. The elements of each weight vector are created by a function that approximates a uniform distribution between -1 and 1. By looking at aggregate results over a large number of trials, we discount the effect of specific weight elements and can consider the properties of the architecture. Inputs are sequentially incremented such that all possible input vectors are covered for each trial. Input space can be represented as an n-dimension unit hypercube where n is the number of sensor nodes. Each element of the input vector varies from zero to one. As input space is covered, the node outputs are analyzed to extract information about the property of interest.

The first property examined is the capacity of a given architecture to represent objects in the neural category. The number of distinct combinations of neural objects that we can represent is a bound on the number of distinct concepts we can represent. We count the number of distinct patterns of discretized node outputs observed as the network input covers input space. Each output pattern is a Γ, i.e. set of signal paths. Though we don't yet put any constraints on how those signal paths relate. Note that as this is a simulation, all values are discretized at some level. Our artificial neural network evaluations are carried out at ANSI C float precision. When we mention a discretized value we have discretized a float value into some much small set of possible values.

Next, the commutative diagram capacity of an architecture is examined. The constraints [62] that a network must meet over its input space in order to be capable

of representing a commutative diagram are enumerated in the Background section. We determine the commutative diagram capacity of two network architectures by searching over random weight vectors for the satisfaction of those constraints. In this context, commutative diagram capacity can be thought of as how well a neural architecture lends itself to representing a commutative diagram from a neural category (and through a functor, the category **Concept**).

For each experiment a network architecture is selected. The architecture will remain constant throughout the experiment. Figure A.4 shows each architecture and the label given to that architecture. Each experiment consists of 100 trials. A new set of connection weights is assigned to the network for each trial. The elements of each weight vector are uniform randomly distributed between -1 and 1. We also select an output discretization size for each experiment. Weight values, activation values, and output values are all stored to float precision. Input-space is covered to a fixed discretization size for the whole experiment. After a network has stabilized, outputs are binned to the output discretization size. If the output discretization size is 0.25, then output values would be classified as $0, 1, 2, 3$ based on the output being in $[0, 0.25), [0.25, 0.5), [0.5, 0.75), [0.75, 1.0]$ respectively. In all cases, the output function of each node is a sigmoid generated by $1/(1 + e^{-\theta})$ where θ is the node activation value.

The first set of experiments counts unique output patterns. An output pattern is an N component vector of the discrete intervals into which the output of each of the N nodes in the architecture fall. The experiment is run using each of the architectures in figure A.4. For the 2x2 and 2x3 architectures, we compare the number of distinct output vectors given output discretization sizes of 0.25, 0.125, 0.0625, and 0.03125. For the 3x2, 3x3 and 4x2 architectures, we use output discretization sizes of 0.25, 0.125, and 0.0625. The 2x2 and 2x3 experiments are run with an input discretization size of 0.015625. The remaining experiments are run with an input discretization size of 0.0625. Each input element takes on either 17 or 65 equally spaced values from 0 to

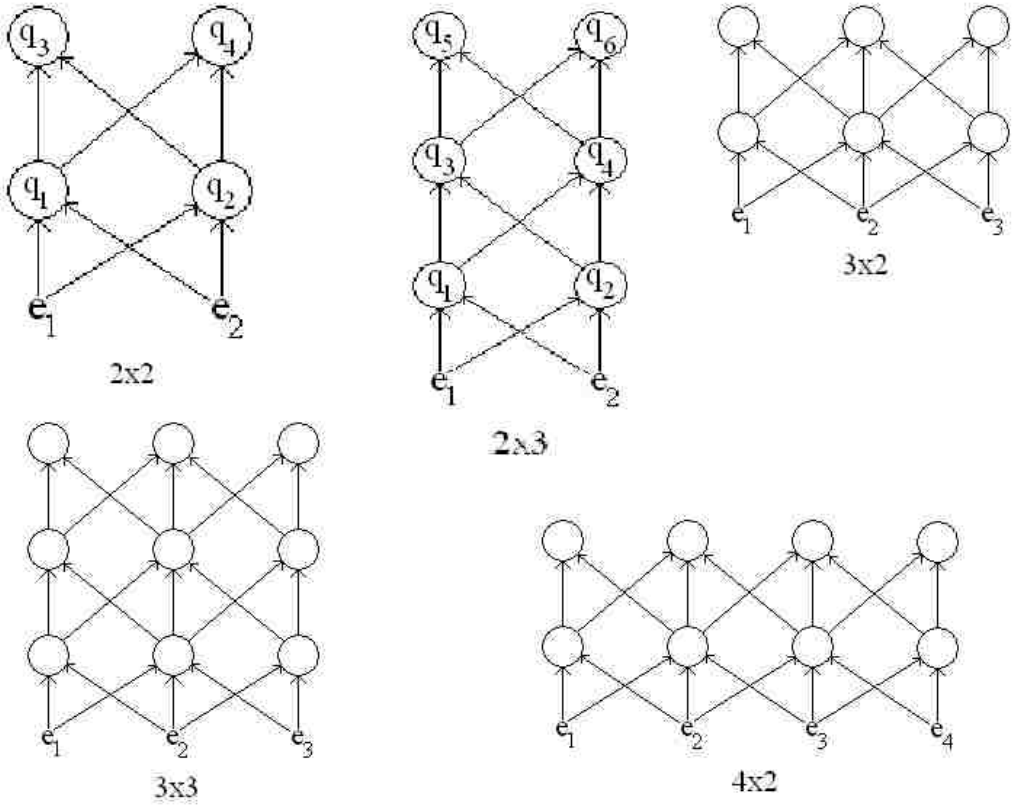

Figure A.4: Neural network architectures used in experiments.

1. Where n is the number of input nodes, we picture an n-dimensional grid of equally spaced points. We increment over all points in the grid and thereby cover input-space. For each input vector we evaluate the neural network repeatedly until the changes caused by the new input vector have propagated to the highest level of the network. A neural network evaluation step is, for each node in the network, the calculation of the node output function on the weighted sum of all input connections to the node. We use eLoom [96], an open source neural network simulation tool, to carry out network evaluation. At this point the network will be stable and exhibit no further changes until we again change the input vector. Recall that the definition of a neural object is a neural network node which has output value within some interval η over

a network transition. The node outputs of our feed-forward networks remain stable indefinitely after an the input values propagate throughout the network. Therefore, the neural objects being represented are constant after that propagation period, until the input vector is changed again.

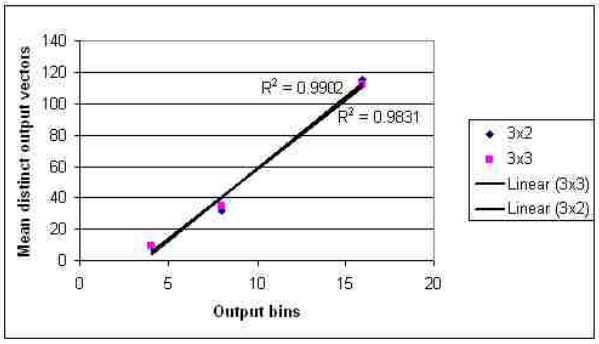

Figure A.5: Mean distinct outputs versus Output bins for 3x2 and 3x3 architectures.

The mean number of unique output patterns is plotted versus the number of output bins. The number of output bins is the reciprocal of the output discretization size plus one. I.e. the number of intervals of a node output that each map to a single discrete value. A line is fit, and R^2 provided, to highlight the data trend. Note that R^2 is a statistical measure of how close a set of data points is to fitting a line. The closer R^2 is to unity, the better the linear fit. The graphs for the 2x2, 2x3 and 4x2 architectures are similar to figure A.5 except with R^2 of 0.9904, 0.9717, and 0.962 to a linear fit. The mean number of unique output patterns is calculated over 100 random sets of network weights for each output discretization size for each architecture. We also consider the histograms of number of distinct output patterns, for each input vector over the 100 trials for a given architecture, see figure A.6.

Next network architecture, weight vector pairs are analyzed to determine if they can represent a commutative diagram. For each architecture, we iterate over 100 random weight vectors. We record the number of weight vectors which allow the neural network to represent a commutative diagram. Ability to represent a commutative

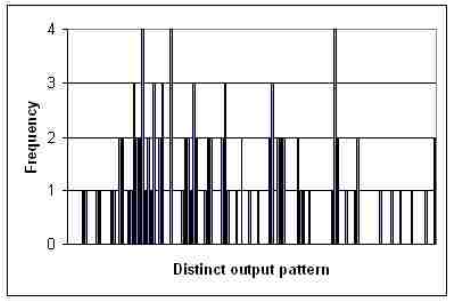

Figure A.6: Histogram of number of distinct output patterns for output discretization of 0.03125 on 2x3 architecture.

diagram is determined by recording the regions over which each node of the network is active. Recall that a node is active if its output is within a predefined interval. We map the active output areas onto input space and test for the required nesting conditions seen in figure A.2. As the input region over which a node will be active is dependent on the interval which defines node activation, we test for a commutative diagram on each architecture, weight vector pair with the active interval defined sequentially as each possible discrete output. If the architecture, weight vector pair is found to be capable of representing a commutative diagram for any of the interval definitions then the count is incremented and we move to the next weight vector on the same architecture.

Our first step in finding commutative diagrams is to look for incomplete but potential commutative diagrams. We consider the 2x2 network architecture. The 2x2 architecture can not represent a commutative diagram because it does not have a third layer to hold the target object. We look for only a source object and two intermediate objects. q_1 and q_2, see figure A.4, are both potential source objects, we first consider the beginning of a commutative diagram using q_1 as source. All rules for later using q_2 as the source are a simple substitution of q_1 and its activation regions for q_1 and its activation regions. Recall that the activation region of q_i is the region of input space for which q_1 is in the interval η_i. In terms of our activation region criteria, we look for (q_3, η_3) and (q_4, η_4) active inside the region of activation

of (q_1, η_1). In addition, the regions of activation for (q_3, η_3) and (q_4, η_4) must have a nonempty intersection. The nonempty intersection is so that if we completed the diagram with a target object, there would be some area where both (q_3, η_3) and (q_4, η_4) could have a morphism to the same target object.

The first part of the commutative diagram testing algorithm consists of three nested loops. The outer loop iterates through possible intervals η_1. The next loop iterates through possible intervals η_2. The third loop iterates through possible intervals η_3. In this way, all combinations of intervals are tested for a partial commutative diagram. Inside of the three interval loops, we iterate over input-space. For each 3-tuple of η intervals, there are five conditions which must hold over all input-space in order for a potential commutative diagram to be represented. If we examine figure A.2, we see these conditions in the venn diagram. Regions illustrated in figure A.2 are the active regions of their respective nodes. Recall that a node p_i is active relative to the neural object (p_i, η_i) if its output falls in the interval η_i. For the 2x2 architecture of figure A.4, the region of input space for which q_3 is active must not extend outside the region of input space for which q_1 is active. q_4 must not be active outside the region of activation of q_1. q_3 must be active somewhere in input space. q_4 must be active somewhere in input space. The activation regions for q_3 and q_4 must intersect for some region of input-space. Implicit in those five conditions is that q_1 be active somewhere in the input space. If all five conditions are true for any 3-tuple of η intervals for a given weight vector, then that weight vector, on the architecture under analysis, is capable of representing the beginning of a commutative diagram. After testing with q_1, q_3, and q_4 we repeat the test with q_2, q_3, and q_4. If any of the eta 3-tuples, for either the q_1 or the q_2 case, reveal a potential commutative diagram then the current weight state is considered capable of supporting a commutative diagram. We count the number of weight states, i.e. trials, that are capable of supporting commutative diagrams out of 100 trials, see figure A.7.

output discretization size	commutative diagrams
0.25	32
0.125	4
0.0625	3

Figure A.7: Number of random weight vectors, out of 100, which could represent a commutative diagram on a 2x2 architecture.

After searching 2x2 networks for possible commutative diagrams, we extend the technique to the 2x3 architecture and possible complete commutative diagrams. Conditions are the same as in the 2x2 case except that now we also have a target node. In the 2x3 case, the target node can be either q_5 or q_6. For each weight state, we first consider q_5 the target node, then q_6. Taking into account the two possible source nodes, we consider in order, q_1 to q_5, q_2 to q_5, q_1 to q_6, and q_2 to q_6. In each case, morphisms must pass through q_3 and q_4 in order to get the two equivalent paths which are necessary in the definition of a commutative diagram. Taking q_5 as the target node, the additional commutative diagram test condition is that (q_5, η_5) exists inside the intersection of the active regions for (q_3, η_3) and (q_4, η_4). As before, if any combination of possible target and source nodes reveals a commutative diagram then the trial is capable of supporting a commutative diagram. In figure A.8, we see the number of trials, out of 100, capable of supporting a commutative diagram.

output discretization size	commutative diagrams
0.25	27
0.125	9
0.0625	1

Figure A.8: Number of random weight vectors, out of 100, which could represent a commutative diagram on a 2x3 architecture.

In figure A.5 we notice that the number of distinct output patterns scales linearly

with the output discretization size. Note that the input discretization size stays constant for a given architecture. The increase in distinct output patterns is not due to an increase in distinct inputs. Output space is an n-dimensional space where n is the number of nodes in the network and the component of the output pattern in each dimension can take on any of the bin values allowed by the output discretization (i.e. 0,1,2,3 if the discretization is 0.25). Had the output patterns been distributed uniformly through output space, then halving output discretization size would result in 2^n time more output patterns than seen at the prior discretization. To see this, consider the case of maximum discretization size with only one output bin. All output patterns look alike; therefore there is only one distinct pattern. Now consider discretization such that there are two output bins. If output patterns are distributed uniformly around output space, then an equal number of output patterns should fall into each of the two output bins in n dimensions, yielding 2^n distinct output patterns. Double the number of output bins again and there are four output bins per dimension for a total of $2^n * 2^n$ possible distinct output patterns. As long as the number of system inputs does not constrain the number of distinct outputs, then for uniformly distributed output patterns we should see exponential growth in the number of distinct output patterns as we exponentially decrease output discretization (reducing discretization size by half at each step). As it is, we only see linear growth in the number of distinct output patterns. There is some compression of input vectors into areas of output space. This follows conceptually in that for a given trained network there should be the ability for multiple sensor observations (i.e. network inputs) to activate the same concept (i.e. output pattern). E.g. two different perspective views of the same object.

A network instance is the state of all node outputs in the network, given a constant weight state and architecture. Each distinct output pattern represents a unique instance of the network. Each unique network instance represents a unique collection of neural objects. The number of unique combinations of neural objects that a

network can represent determines the concept capacity of that network.

The relatively large spread of frequencies in the number of distinct output pattern histograms, see figure A.6, presents an interesting bit of information. The number of distinct output patterns is an upper bound on the number of distinct concepts. If the number of distinct output patterns realizable by a network is highly variable upon the weight state of the network, then the number of concepts that a network can represent is a function of the connection weights of that network. This result only explicitly holds true for the fully connected feedforward networks examined in this work. However, as many, more complex networks will likely have a large number of feedforward elements, it seems reasonable that this result will be present to some degree in other networks. Extrapolating to trained neural networks, this result suggests that if we want a neural network to be able to represent a large number of concepts then we will have to take care in the network's training such that its connection weights end up in a state which allows a large number of concepts. Some areas of weight-space may only allow a relatively small number of concepts to be represented by the network. To meaningfully specify the concept capacity of a network, we need to know not only the architecture but also the current weight state.

Figure A.8 shows that the ability of a network to represent a commutative diagram scales in a reasonable manner with output discretization. As we go to smaller output discretization, the active region of each node, for any given η, will become smaller because there will be more possible ηs for each node to fall into. Consider the nested Venn diagram of figure A.2. If the regions become smaller on average, then it will be harder for random weight vectors to induce the desired nesting of regions.

The result from counting distinct output patterns suggests that simply increasing the number of neural objects per neural network node may not be a sufficient way to increase concept storage. Additionally, the commutative diagram results suggest that we will want to develop training guidelines to maximize the conceptual capacity

of a given network.

A.5 Theoretical Bounds on Categorical Concept Capacity

Following the investigation of applied methods, a combinatorics based theoretical method was developed for the same type of networks. Numerical work in the previous section calculated average commutative diagram upper bounds for a given architecture. In this investigation, we take those same fully connected by layer feedforward architectures and find a theoretical upper bound on their capacity to represent commutative diagrams. This theoretical upper bound is based only on the structure of the neural network and not on specific implementations as in the first set of experiments.

The second investigation involved a combinatoric count of possible commutative diagrams. Counting possible commutative diagrams gave average commutative diagram upper bounds (i.e. capacity) for given architectures with specific weights. Now, those same fully connected by layer feedforward architectures are analyzed for a theoretical upper bound on their capacity to represent commutative diagrams. This theoretical upper bound is based only on the structure of the neural network and not on specific implementations as in the first set of experiments.

Neural networks derive meaning from the relative connections between nodes. The structure of a given network is specified by these relative connections. Category theory is the mathematical discipline that is concerned with structure. As category theory provides a rigorous framework to analyze structure, it provides a rigorous method to analyze those properties of neural networks that are determined by structure.

We postulate that a natural mathematical context for the operation of neural networks is representation and manipulation of the structure of input information. In

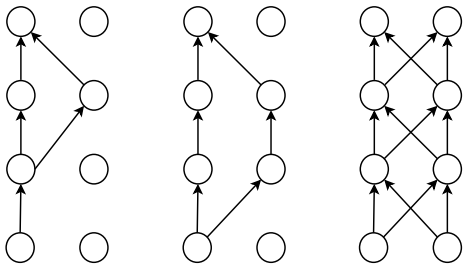

Figure A.9: Commutative diagram path examples. The left panel illustrates one possible commutative diagram over these node, and the center a different possible commutative diagram. The right panel illustrates all possible paths using the fully connected by layer scheme.

the case of biological, or otherwise embodied, neural networks, the input information is supplied by sensors reacting to environmental observation. Letting a concept be any element of information, the structure of input information is the conceptual structure of the observed environment. The Categorical Neural Semantic Theory (CNST) by which an individual concept can be represented as an object in the *Concept* category, and the activation of a neural node within a certain interval, is detailed in the work of Healy and Caudell [62]. These ideas of object and category are two fundamental ideas of category theory. An overview of category theory is found in A.

A commutative diagram requires two or more paths with common source and target nodes. Each node in the path may be one of a number of neural objects depending upon what interval its activation falls in. A unique instance of neural objects and connections, that form two or more signal paths between common source and target nodes, is a unique commutative diagram in the neural category.

By counting the number of unique instances of two or more signal paths between each possible source and target node, we calculate an upper bound on the number of unique commutative diagrams representable by a given architecture with a given output discretization size and object forming interval scheme. We begin with fully connected by layer, feedforward, architectures as used in [4].

Consider the example architecture of figure A.9. The right panel shows all archi-

tecture connections. The center panel shows one way to form a commutative diagram with this architecture. The left panel shows another way to form a commutative diagram. For a given set of paths, a unique commutative diagram is formed for each unique combination of neural objects used in the paths. We will count all possible ways to form commutative diagrams in this architecture, then in other architectures of the fully connected by layer feedforward form.

First, fix the target and source nodes. Paths from source to target must then choose one of the two nodes at each of the two middle layers. A set of two or more valid paths forms a commutative diagram. In this case, $\binom{2}{1} * \binom{2}{1} = 4$ different paths can be formed given that there are 2 nodes in each layer and 1 node is chosen from each layer. Given four possible paths, we can form $\binom{4}{2} + \binom{4}{3} + \binom{4}{4} = \sum_{k=2}^{4} \binom{4}{k} = 14$ unique, commutative diagram forming, path sets. Note that a commutative diagram forming path set needs include at least two paths and may include up to all paths from source to target. We now account for the selection of any possible source-target pair by noting that there are two choices for the source and two choices for the target. This gives us $2 * 2 \sum_{k=2}^{4} \binom{4}{k}$ commutative diagram forming path sets through this architecture. The last step is to consider the actual neural objects forming each path set. Two topologically identical path sets are distinct if any of the neural objects composing the paths are different. Each set of paths need not include all nodes in the middle layers (middle layers being those which do not contain the source or target layer), but to create an upper bound, we consider the number of distinct neural node configurations formed by the middle layers along with the target and source nodes. For the architecture under consideration, we have 4 nodes in the middle layer plus a target and source node for a total 6 node upper bound on the number of nodes in any given path set. Each node may be one of as many different neural objects as there are distinct intervals defined to form those neural objects. At most, there may be as many distinct object intervals as there are discrete neural output values. We define d as the number of distinct intervals defined to form neural objects. In

this case, we have up to d^6 distinct combinations of neural objects along a set of paths from a source to a target node. Therefore, this architecture can support up to $d^6 * 2 * 2 \sum_{k=2}^{4} \binom{4}{k}$ distinct commutative diagrams.

In a more general case, for any fully connected by layer feedforward network with each layer the same size, we can specify a network by its length l and width w. For each path, we pick one node from w choices for each of the $l-2$ middle layers resulting in w^{l-2} distinct paths from any given source node to any given target node. Forming path sets from those paths, we can make $\sum_{k=2}^{w^{l-2}} \binom{w^{l-2}}{k}$ distinct path sets. Adding the consideration of distinct neural objects, each path in a set can contain at most $w * (l-2)$ neural objects. This yield as upper bound of $d^{w*(l-2)} * \sum_{k=2}^{w^{l-2}} \binom{w^{l-2}}{k}$ distinct potential commutative diagrams from any given source to any given target node. Therefore we have $w * w * d^{w*(l-2)} * \sum_{k=2}^{w^{l-2}} \binom{w^{l-2}}{k}$ distinct potential commutative diagrams from arbitrary first level source to arbitrary last level target.

A.6 Categorical Encoding of Conceptual Structure

The theory that brain operation, at the conceptual level, can be expressed as a function of structure is illustrated in Figure A.10. The lower left quadrant represents afferent sensory information from the world. It could be said that the quadrant actually represents the world as it exists external to the brain, but then we run into philosophical objections as to whether there is any world outside the brain. As such, we only claim there is some information that appears to stimulate an organism's senses. For simplicity's sake though, we will henceforth refer to the world as if it is an external reference. The lower right quadrant represents neural tissue. The upper left quadrant represents the conceptual structure resulting from a given instance of the world. The upper right quadrant represents the conceptual structure resulting from a given instance of neural tissue. The main idea of the structure based theory

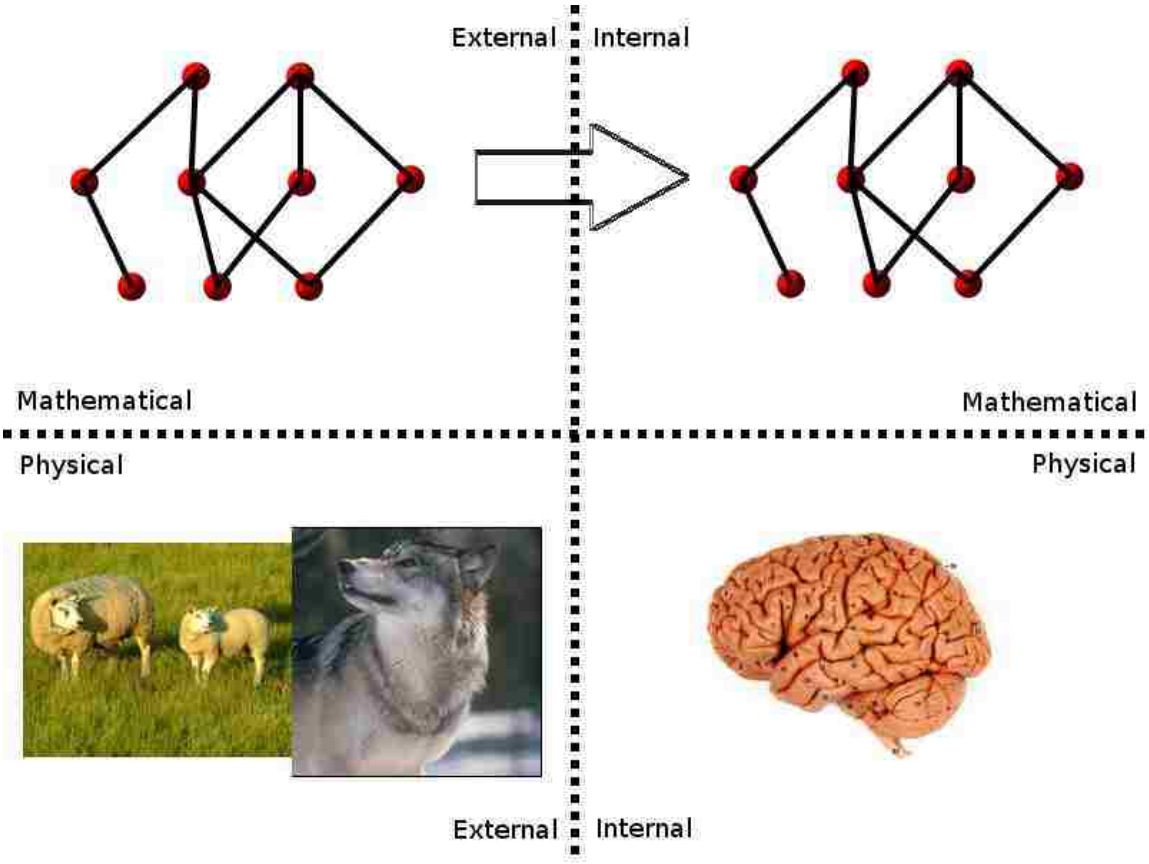

Figure A.10: Structure of categories relating brain and environment.

is that the most fit organisms will have a structurally invariant mapping between the conceptual structure of the world and the conceptual structure of their neural tissue because this will allow neural processing to produce conclusions about the external world. As our goal is to embody neural networks that invariantly map conceptual structure from the world to the artificial neural tissue, evidence will be produced either for or against the structure based theory of brain operation.

We assert that biological neural tissue is able to represent the structure of the physical world. I.e. when we perform an inflective task of rotating an object in our imagination, we are performing operations upon neural tissue activation levels where the neural tissue activation levels map to an object of the physical world, and operations on the activation levels map to physical rotation of the object. Given that

assertion, there will exist a mapping from physical objects to neural activations in a particular instance of neural tissue. In order to be mathematically consistent, that mapping must satisfy certain rules described by category theory [110]. A category must have definitions for objects and morphisms between those objects. A category must satisfy two "bookkeeping" rules, as well as the identity rule and the associative rule. The first bookkeeping rule is that the domain and codomain of I_A are both A. Where I is the identity morphism. The second bookkeeping rule is that $g \circ f$ is only defined if the domain of g is the codomain of f, the domain of $g \circ f$ is the domain of f and the codomain of $g \circ f$ is the codomain of g. The identity rule requires that if $A \xrightarrow{f} B$ then $I_B \circ f = f$ and $f \circ I_A = f$. The associative rule requires that if $A \xrightarrow{f} B \xrightarrow{g} C \xrightarrow{h} D$ then $h \circ (g \circ f) = (h \circ g) \circ f$.

The fact that there are rules for valid categories, and that valid categories are necessary to perform valid structure mappings suggests that instantiations of neural systems will have to meet certain rules to perform optimally. One of the contributions of this dissertation is to demonstrate that addition of a categorically valid neural structure improves the performance of the embodiment of that system.

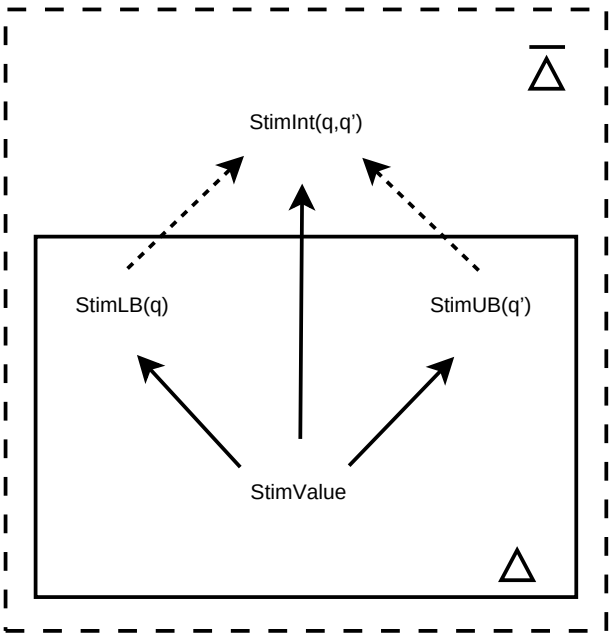

Figure A.11: A stimulus concept colimit with base diagram Δ and defining diagram $\bar{\Delta}$. The apical object $q < Activation < q'$ could also be represented as a stimulus interval object $StimInt(q, q')$.

Appendix B

Categorical Limits

This appendix describes an ART variant which has the power to represent categorical limits and colimits. Categorical limits and colimits represent generalization and specificity in a neural architecture.

B.1 LimitsART Implements Limits and Colimits

The method which will be implemented to allow limit and colimit representation corresponds to control over the length of each side of template hyperboxes. This subsection specifies how that method is implemented and why it corresponds to categorical limit and colimit representation.

The neural structure which controls hyperbox length is the ρ vigilance node in conjunction with the resonance interface between the $F1$ layer and the $F2$ layer. Controlling the maximum size of each hyperbox side means controlling the individual regions T_d^k of each template T^k to restrict the maximum difference among all possible values $r = q' - q$. The strategy taken is adapted for fuzzy ART from [115]; in the segment of the $F1$ layer for each stack interval network, attach the architectural structure necessary to represent the colimit apical object $StimInt(q, q']$ (shown in Figure A.11) to each pair q, q' of $F1$ nodes having the desired difference $r_{max} = q' - q$

between their neural activations.

B.1.1 The Colimit Representations

Consider the ART network $F0$ and $F1$ layers subdivided into n segments. These segments might be labeled pattern components, hyperbox dimensions, feature elements, or something else appropriate for a particular input vector context. The discussion in this section will apply separately for each value of d, where $1 \leq d \leq n$. All n segments will be considered together when appropriate.

The concepts $StimLB_q$ and $StimUB_{q'}$, their common subconcept $StimVal$, and the morphisms $StimVal \rightarrow StimLB_q$ and $StimVal \rightarrow StimUB_{q'}$ in the base diagram Δ of the desired colimit (Figure A.11) are already explicitly represented by the activation of segment d of $F0$, and hence, $F1$. To represent the colimit, add a node and connections representing the colimit apical object $StimInt(q, q']$ and its cocone leg morphisms, whose domains are base diagram objects (reciprocal connections are also needed, since here it is necessary to actively enforce the model-space morphism property). The leg morphisms $StimUB_q \rightarrow StimInt(q, q']$ and $StimLB_{q'} \rightarrow StimInt(q, q']$ are represented by additive connections from the two $F1$ nodes representing $StimLB_{q'}$ and $StimUB_q$ together with the reciprocal connections. The leg morphism $StimVal \rightarrow StimInt(q, q']$ is represented by the set containing the two paths from the $StimVal$ node, through the nodes representing $StimLB_{q'}$ and $StimUB_q$, and terminating at the $StimInt(q, q']$ node, along with a reciprocal connection from the $StimInt(q, q']$ node to the $StimVal$ node. Small, positive fixed weights for these connections ensure that the colimit node requires input from both nodes q and q' in order to become active, thereby ensuring that one path representing the colimit defining diagram is active if and only if the other path is as well. This ensures the commutativity of the defining diagram for the colimit in the category $N_{A,w}$, which represents the defining diagram for the $StimInt(q, q']$ colimit in the category **Concept**.

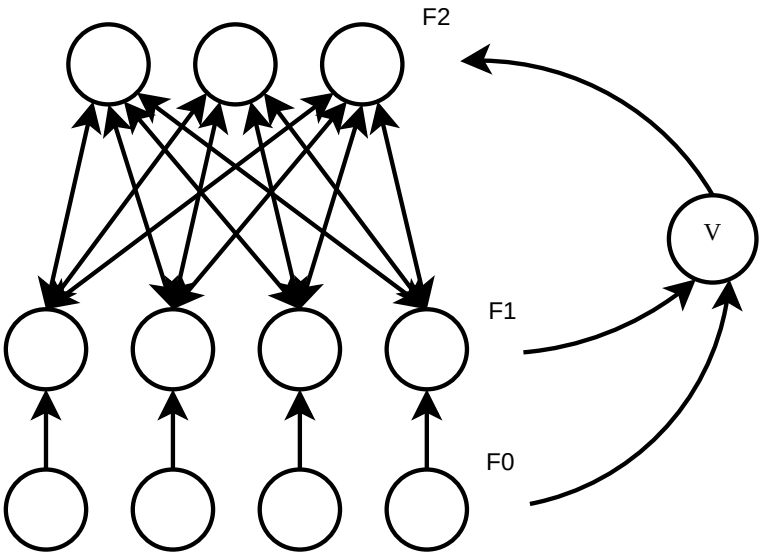

Figure B.1: Unmodified ART illustrated here for comparison to LimitsART.

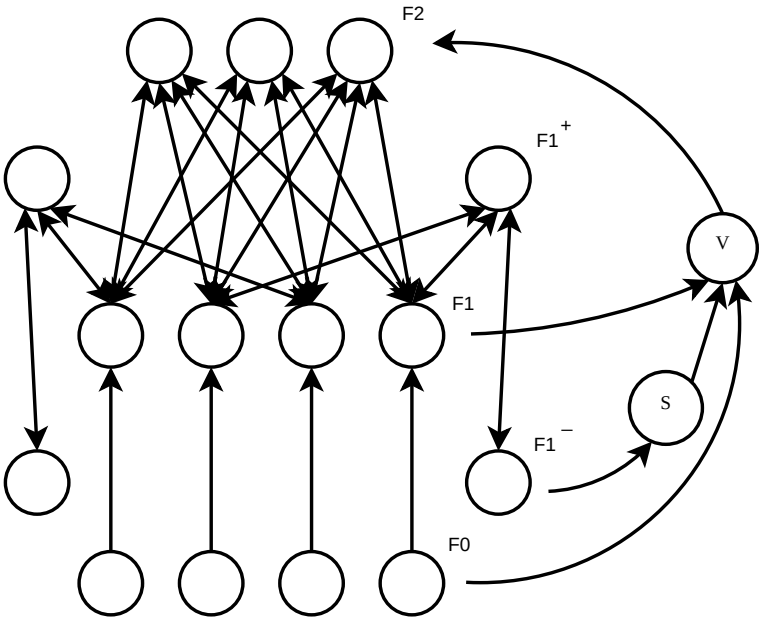

Figure B.2: LimitsART augments the $F1$ layer with colimits and limits.

Figure B.2 shows two colimit apical object nodes and connections added to each input dimension of the fuzzy ART network of Figure B.1. In the case of Figure B.2, there are two input dimensions in the input vector. The $F0/F1$ nodes are, in sequence, the two real valued inputs followed by the complements of those inputs.

The complement of each real valued input r is $r_{max} - r$, where r_{max} is the maximum node activation value. Each feature dimension has a colimit apical object node. The nodes representing the colimit apical objects $StimInt(q, q']$ for each of the n input dimensions all together form a new layer which is called $F1^+$.

Suppose that the maximum hyperbox side length to be allowed is r_{max}. This determines a range of real valued pairs (q, q') with $q \leq N - r_{max}$ (where all quantities are as described in the previous paragraphs) and $q' = max(q, 0) + r_{max}$. These pairs represent all possible intervals having the maximum length r_{max} for side d of any template hyperbox, corresponding to an interval $q < v \leq q + r_{max}$. The maximum side length r_{max} is to be applied uniformly across the n dimensions.

To understand why the colimits represent maximum side lengths, suppose that a hyperbox "grows" so that side d has length r and is represented by the pair $(q1, q2)$ with $q2 = q1 + r$. Suppose further that the side exceeds the maximum length, $r > r_{max}$. This means that when the corresponding template is active, and for any of colimit pair (q, q') from the possible range, either node q is inactive (if $q_1 < q$) or node q' is inactive (if $q' < q_2$). Then, none of the colimit nodes in that segment of $F1^+$ can remain active since at least one of its leg morphisms must be inactive, the nodes q, q' being the sources of the connections implementing those leg morphisms (refer to Figure B.2).

B.1.2 The Limit Representations

There is a range of $StimInt(q, q']$ concept colimit representations for each hyperbox dimension d. However, it is irrelevant to our purpose which interval in this range contains the actual hyperbox side d: only the side length r_{max} need be considered, since that is what we wish to control. This control can be exerted by applying the knowledge that exceeding the side length will result in all colimit nodes becoming inactive in segment d of $F1^+$. This calls for an abstraction: remove the information specific to each particular pair (q, q'), leaving only their common side length, which

is to be compared with r_{max}. Since a particular pair is represented by a colimit node, the abstraction can be achieved by attaching the node and connections necessary to form a limit cone for a diagram containing desired colimit apical objects represented in segment d of $F1^+$, including reciprocal connections for the limit leg morphisms(see Figure B.2). To complete the limit defining diagram, an upper level node is added representing a common concept that the colimits all map to. Otherwise, the diagram is discrete, and there are virtually no limits for discrete diagrams in the concept category.

In [115], we see that for stack coded binary ART1 there are n stack interval networks which correspond to n limit cones. For real valued fuzzy ART, intervals defined by (q, q') can fall anywhere in a continuous range, so certain intervals must be chosen at implementation time to be represented by actual nodes. As shown in Figure B.2, apical object nodes form a new layer designated $F1^-$, which has a single node for each of the n segments of $F1^+$. The added node L is in all the base diagrams, it represents a concept $AllStimInt$ formed from $StimVal$ and the concepts $StimInt(q, q']$ via concept morphisms represented by the appropriate connections. Its inclusion is necessary for the limit cones to correctly represent concept limits. If $r_{max} \leq \frac{1}{2}N - 1$, $AllStimInt$ will be an inconsistent concept, since it requires the activation value v to be in all intervals at once. As is described for the original stack coded theory, inconsistent theory representations cause no harm as long as they are not given a role in network operation, and the only role for node L is to represent $AllStimInt$. The $F1^-$ nodes and their attendant connections correspond to limit cones in the neural category $N_{A,w}$ which are the images of the appropriate limit cones in **Concept**.

Using the property of a functor M : **Concept** $\rightarrow N_{A,w}$ and other categorical principles, we have redesigned the ART network to obtain LimitsART, which enforces the properties conferred by colimits and limits from the category concept. As long as some $F1^+$ colimit object node is active in each segment d, its limit object node

in F_1^+ will be active because of the model-space morphism property designed into the connections representing the limit cone morphisms. Since it has no other inputs, the limit node will be inactive if all colimit nodes are inactive. Now, as previously shown, the requirement that some colimit node in segment d be active is $r \leq r_{max}$, which is to say that the interval (q, q') represented by a given hyperbox side must not exceed the maximum specified interval.

B.1.3 Limit Representations Control Vigilance

The following describes how fuzzy LimitsART implements the effects of limits upon template formation. This implementation description is adapted from the Neurocomputing paper which describes LimitsART in detail [115]. Connections from the $F-1^+$ nodes to the ART vigilance node V through the node labeled S as shown in Figure B.2 allow each $F-1^+$ node to supplement the vigilance node's $F2$ reset capability, as follows. First, node S is tetanically active except when the entire $F-1^+$ layer,acting cumulatively, suppresses its activation through n inhibitory(-) connections targeting S, one for each $F-1^+$ node. Therefore, if resonance between the current input and a template pattern is about to occur, but any one of the $F-1^+$ nodes is inactive (because all of its $F1^+$ correspondents are inactive), the resulting lowering of the cumulative inhibitory signal to S allows it to become active. Acting through its additive connection to the vigilance node V, S thereby activates V, effectively vetoing the resonance via an $F2$ layer reset.

To avoid activating V each segment d must satisfy the inequality $\left\|I_d \wedge T_d^k\right\| \geq N - r_{max} = t \cdot N$. An $F1$ activity pattern $I \wedge T^k$ is made up of the activity patterns $I_d \wedge T_d^k$. The added network structure in LimitsART (Figure B.2) provides for a uniform upper bound on hyperbox side length. This allows the user of the network to exercise a more specific control over template information "erosion" during recoding than is allowed by having a vigilance parameter alone. Just as ρ can be applied to control the amount of specialization versus generalization allowed in the

templates, t can be applied to control the specialization versus generalization allowed in each of the n segments of the templates. A higher value for either parameter means smaller hyperboxes, hence fewer input exemplars per template, and therefore greater specialization.

Just as in the stack coded case, it might be considered whether the parameter ρ can be eliminated altogether. I.e. does t make ρ redundant? It can be shown that the usual test involving ρ is indeed redundant if $t \geq \rho$. However, this is not the case when $t < \rho$,and therefore the parameter ρ is not redundant.

B.1.4 Apparatus

In this section, an experiment is detailed that implements LimitsART and is designed to test whether LimitsART provides an improved classification of input data when compared with that obtained with unaugmented ART. The experiment is carried out in a standalone context, i.e. not as part of any greater neural architecture. The purpose of the experiment is to provide evidence as to whether it might improve performance to incorporate LimitsART into the larger embodied architecture.

The data domain for the experiment was chosen to enable tests of separate hypotheses about performance improvement. A single LimitsART simulation program was used. The simulator was programmed so that with a nonzero value of the aforementioned threshold value t it simulates LimitsART at that threshold, and with $t = 0$ it simulates ART.

The LimitsART simulator was produced using the network specification and simulation package eLoom [96]. A major advantage of working with variations on ART is that, while the theory behind it is substantial and mathematically guarantees that it performs classification in a well-defined manner for items represented by binary input patterns [63], its simulation involves a relatively simple algorithm. LimitsART simulation is equally simple. This simplicity can be attributed directly to the mathematical performance guarantees of the theories behind ART [65] and LimitsART

[62]. In particular, a choice of colimits, all representing intervals with the same length for all segments of $F1$, determines the maximum allowed hyperbox side length. The choice is specified as the spacing of nodes in the colimit base diagrams, given by the threshold parameter t for the activation of limit apical nodes. Thus, an abstraction derived from the semantic theory (limits built upon colimits) has given rise to a numerical parameter. This is to say, the theoretical consideration of category theory has resulted in an implementable design element.

B.1.5 Experiment

The results of the stand alone LimitsART experiment can be found in appendix E. The conclusion based on experimental results is that LimitsART was better able to form templates which efficiently covered the input data. Unaugmented ART was able to learn classifications of all input data, but ART formed templates spanning over large regions of input space in which no inputs occured. In the sense that classification templates are generalizations of groups of input points, LimitsART performs better at the generalization task because it only formed templates around regions in which input points occure. Based on this performance, LimitsART will be incorporated into the embodied architecture to see if the improvements derived from category theory will improve embodied performance.

Appendix C

Extended Background Material

C.1 Hippocampus

In addition to activation during item and context association, the hippocampus is also selectively activated during the encoding or retrieval of verbal [116] [117] and face-name associations [118] [119] [120]. Correspondingly, recent neuropsychological studies have found that recognition of associations is impaired even when recognition for single items is spared in amnesic patients [121] [122]. These studies reported impairment in recognition memory for associations between words or between faces or face-occupations pairs, as compared to normal performance in recognition of single items. At the same time, other functional imaging studies and characterizations of amnesia have suggested that the hippocampus is sometimes involved in both associative and single item recognition, highlighting the need to clarify the nature of associative information that composes an "event" [123]. To build evidence that a successfully model of hippocampus which displays some of the behavior demonstrated by the preceding literature can be computationally realized with the methods presented in the dissertation research, a computational hippocampus model will be presented.

Many component elements combine to create context-rich episodic memories.

Functional imaging studies provide evidence to support activation of the hippocampus during the encoding or retrieval of associations among many elements of a memory [124] [125] [126] [127]. For example, Henke [128] observed greater hippocampal activation when subjects associated a person with a house, as compared to making independent judgments about the person and house. Others have found selective hippocampal activation during recollection of the context in which prior learning occurred while performing formal tests of memory [129] [130].

The coding of associations has been experimentally demonstrated to extend beyond item and context associations. These findings are generally consistent with the notion that the hippocampus plays a distinct role in recollection associated with binding features of items and their context to represent salient events [131]. The environment in which the architecture of this dissertation is embodied will by nature be limited to the sensory stimuli which encode the distal reward task. Similarly, there is no temporal existence other than during the task. As such, all of the embodied architecture's existence is salient and Eichenbaum's results are informative to the embodied architecture. Given that, evidence present in the Eichenbaum study suggests the hippocampal function binds recollection with item features and their context.

Furthermore, these sequential codings can be envisioned to represent the series of events and their places that compose a meaningful episode. The information contained in these representations distinguishes related episodes that share common events and therefore could be confused. Several studies on the spatial firing patterns of hippocampal neurons as animals traverse different routes that share overlapping locations provide compelling data consistent with this characterization. In one study, rats were trained on the classic spatial alternation task in a modified T-maze [132] [133] [35]. Performance on this task requires that the animal distinguish left-turn and right-turn episodes that overlap for a common segment of the maze and requires the animal to remember the immediately preceding episode to guide the choice on

the current trial, and in that way, the task is similar in demands to those of episodic memory. If hippocampal neurons encode each sequential behavioral event and its locus within one type of episode, then most cells should fire only when the rat is performing within either the left-turn or the right-turn type of episode. This should be particularly evident when the rat is on the "stem" of the maze, when the rat traverses the same set of locations on both types of trials. Indeed, a large proportion of cells that fired when the rat was on the maze stem fired differentially on left-turn versus right-turn trials. The majority of cells showed strong selectivity, some firing at over ten times the rate on one trial type, suggesting they were part of the representations of only one type of episode. Other cells fired substantially on both trial types, potentially providing a link between left-turn and right-turn representations by the common places traversed on both trial types.

Functional imaging studies in humans have also revealed hippocampal involvement in both spatial and non-spatial sequence representation. Several studies have shown that the hippocampus is active when people recall routes between specific start points and goals, but not when subjects merely follow a set of cues through space (Hartley et al.2003). In addition, the hippocampus is selectively activated when people learn sequences of pictures (Kumaran & Maguire, 2006). The stand alone hippocampus model which is refined into the hippocampal compnent of the embodied neural architecture of this dissertation is tested with picture sequences. Even greater hippocampal activation is observed when subjects must disambiguate picture sequences that overlap, parallel to the findings on hippocampal cells that disambiguate spatial sequences [132].

Memories are networked to support inference. Further consideration of the cognitive properties of episodic memory suggest that related episodic representations might be integrated with one another to support semantic memory and the ability to generalize and make inferences from memories. Referring to how related memories are integrated with one another, William James [134] emphasized that "in mental

terms, the more other facts a fact is associated with in the mind, the better possession of it our memory retains. Each of its associates becomes a hook to which it hangs, a means by which to fish it up by when sunk beneath the surface. Together they form a network of attachments by which it is woven into the entire tissue of our thought." James envisioned memory as a systematic organization of information wherein the usefulness of memories was determined by how well they are linked together.

There are two main outcomes of the linking of representations of specific experiences. One is a common base of associations that are not dependent on the episodic context in which the information was acquired. Thus, when several experiences share considerable common information, the overlapping elements and common links among them will be reinforced, such that those items and associations become general regularities. The representation of these general regularities constitutes semantic "knowledge" that is not bound to the particular episode or context in which the information was encoded. The networking of episodic memories by common elements provides a mechanism for the commonly (though not universally [135]) held view that semantic knowledge is derived from information repeated within and abstracted from episodic memories.

Eichenbaum has conducted a series of studies to research the involvement of hippocampus in the linking of memories. The role of the hippocampus in making indirect associations was examined in one particular study. This study examined associations between stimuli that were each directly associated with a common stimulus. Initially, normal rats and rats with hippocampal lesions were trained on a series of overlapping "paired associates" [136]. The study procedure consists of a series of trials. On each trial, one of two items were initially presented to the rat in a pairing. The rat then had to select the arbitrarily assigned associate item. For example, given that the rat was trained on the pairs A-B and X-Y, if A was the initial item, then the rat was expected to select B and not Y; conversely, if X was the initial item the rat was expected to select Y and not B. Afterward, the rats were trained on a

second paired associated list where the initial items were the second items in the first list and new items were the associates (B paired with C and Y paired with Z). Thus, when B was presented initially, the rat was expected to select C and not Z. When Y was presented initially, the rat was now expected to select Z and not C. After training on all four paired associates, recall of the indirect relations among the pairings was tested for each rat. These tests were carried out with presentations of an initial item from the first learned paired associates (A or X) followed by a choice between the second items of the later learned associates (C versus Z). Normal rats demonstrated their ability to express these indirect relations by selecting C when A was presented and Z when X was presented, whereas rats with selective hippocampal damage showed no capacity for this inference from memory. These findings, as well as results from another transitive inference task [137], indicate that the hippocampus is critical to binding distinct memories into a relational network that supports flexible memory expression. The embodied architecture is this dissertation creates a temporal and spatial framework among sensory memories.

Appendix D

Computational Cortical Hippocampal Model

D.1 Episodic Memory

Implementing episodic memory requires at least two components; one component to encode the semantic content of sensory input from the observed environment, and another component to encode the trajectory of sensory input through semantic space over time. A cortical model is presented, with biologically plausible function, that can perform temporal semantic encoding. This cortical model is joined to a neuromorphic hippocampal model for the purpose of emulating biological episodic memory function. Experimental evidence is presented to show that the episodic memory architecture is exhibiting biologically equivalent episodic memory behaviors.

D.2 Cortical Model

The cortical model is comprised of stacked layers of fuzzy Adaptive Resonance Theory (ART) networks. While ART is a good choice for the neuro-cortical classification modules, other unsupervised learning techniques could be expected to render similar

results, though probably with subtle and interesting differences. Between each pair of ART modules there is a layer of temporal integrators (Taylor et al., 2009), with an adjustable time constant of integration for each layer. This TIART network models a biological cortical column, which receives afferent connections from either a particular subsection of the input field or a particular subsection of the previous layer. These subsections overlap in the manner of biological cortex (Kingsley, 2000). The system as a whole is a simple but powerful cortical classifier. Progressively higher layers encode progressively more abstract objects or spatial locations. Low levels correspond to simple perceptual primitives (edges etc.), high levels might correspond to whole objects or other semantic concepts. Corresponding to the temporal episode representation method described earlier, the cortical ART networks are modified. Each ART network has a "top down" recall mode driven by input from higher-level, more abstract layers as well as the traditional "bottom up" mode driven by stimulus input. While in "top down" mode, an activated F2 node in the ART network reinstates the prototypical input pattern which it encodes in synaptic weights between layers F1 and F2 (see Figure 4). So, for example if a particular F2 node encoded the concept of "dog" it could read out the features ("fur," "tail," "barks," etc.) in the F1 layer. These features would in turn activate the F2 layer of the next lowest ART module and so on. In this way the network learns new inputs from the bottom up, and can then recall and reconstruct these features from the top down.

D.3 Hippocampal Model

This section presents the detailed computational hippocampus model. Figure D.1 illustrates the computational hippocampal model with approximate neuron counts from human biology noted. Actual neural node counts are not preserved in the computational model, but relative complexity of modules as reflected in ratios of neuron counts is preserved. Experimental results are provided as evidence for biologically

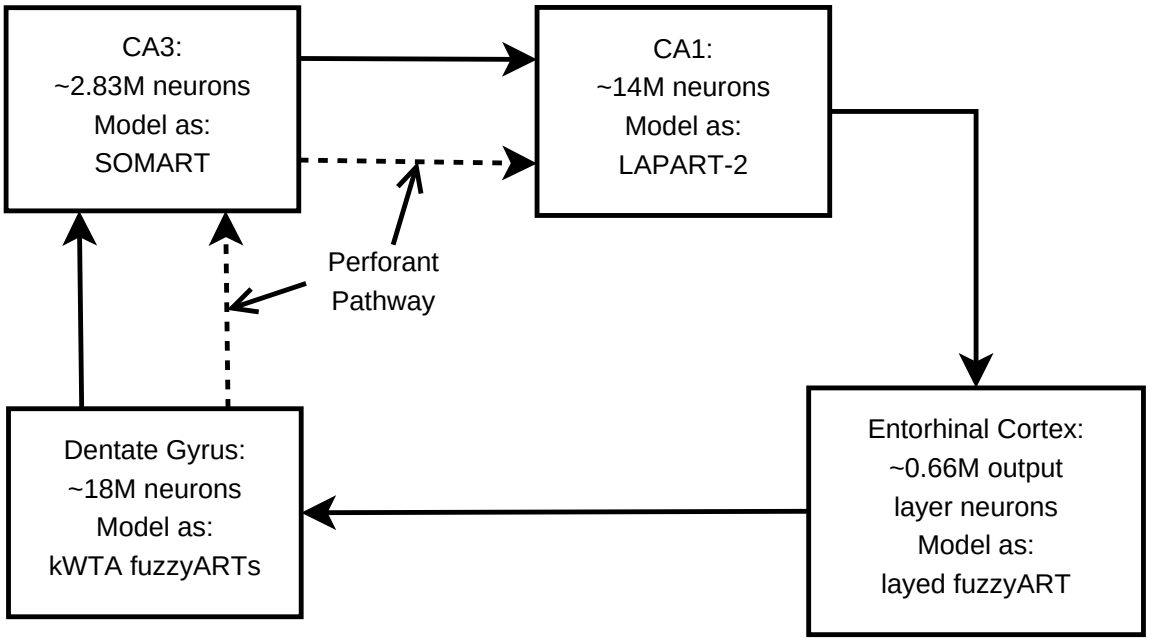

Figure D.1: Hippocampal model with biological neuron counts.

faithful behavior. As was discussed, this stand-alone model will be refined for incorporation into the final embodied architecture. Through computational reproduction and experimentation, understanding is demonstrated of hippocampal structure and function as it relates to episodic memory which is implicated in distal reward tasks. The computational hippocampal model itself is a contribution to the fields of cognitive neuroscience and neuro-psychology. The design of this hippocampal model was undertaken in collaboration with Howard Eichenbaum, Neal Cohen, Michael Bernard, Stephen Verzi, Dan Morrow, Patrick Watson, and Thomas Caudell.

The model of the hippocampal system makes use of several ART variations. It is related to the model of the cortical system, but has a very different architecture and accomplishes fundamentally different information processing. While the cortex attempts to represent the conceptual structure of its inputs, the hippocampus attempts to quickly bind snapshots of high level cortical activity. Behaviorally, this produces an episodic memory mechanism, where concepts originating in multimodal sensory input are bound together. By way of this binding, the hippocampal representation

can also be used to recover neocortical representations from partial activations.

Hippocampus is modeled as a loop of neural modules starting at entorhinal cortex, proceeding to dentate gyrus, continuing to CA3, and then returning to entorhinal cortex through CA1, where some of the function of subiculum is implicitly captured in CA1. Entorhinal cortex is the last level of cortex on the way from sensory input to the hippocampus. Entorhinal cortex is where all information that will be encoded in episodic memory must converge. Dentate gyrus provides a pattern separation function for the information received from entorhinal cortex. CA3 provides pattern completion and semanto-spatial association. CA1 closes the loop and provides temporal association.

D.3.1 Entorhinal Cortex

The entorhinal cortex (EC) is an area of multi-modal convergence [138] where several data streams from different senses come together. The EC's cytoarchitechture resembles that of the cortex, so it here is modeled similarly to the cortical model. However, the ART networks which make up the ERC have two sets of connections to the hippocampus, a feed forward connection to the dentate gyrus and CA3 meant to simulate the perforant path, and bidirectional connections to the CA1/Subiculum component of the hippocampal model. The forward connections provide inputs to the hippocampal module. The back connections use LAPART rules to learn associative links between activity in CA1 and ERC, thereby closing the autoassociative hippocampal loop. When a CA1 representation is activated, these back connections can drive top-down cortical recall.

In our implementation the pre-MTL sensory cortex and EC are represented by layers of fuzzy-ART modules which are modified to encode temporal semantic data. Individually, these temporally integrated adaptive resonance theory (TIART) modules are capable of encoding categorical representations of their given input vectors over time [73]. By combining layers of TIART modules, our EC creates categories

of categories to represent larger semantic concepts and combine the "dorsal stream" containing contextual information and the "ventral stream" of focal information before these streams enter the hippocampus. Within the hippocampal representation in our model, each of the primary regions is represented by a different ART variant selected to achieve the particular functionality of the individual region. The relative size of each module is scaled in accordance with approximate human neuroanatomy (Figure 2).

D.3.2 Dentate Gyrus

The dentate gyrus (DG) has peculiar anatomical properties. It has a large number of neurons with relatively low activity and it is one of the few places in the brain in which new neurons are generated in adults. These properties have lead to the suggestion that the DG creates sparse, non-overlapping codes for unique events via pattern separation [139]. The DG model module is implemented with a layer of k ART networks with large input fields. Each of these networks has a single winner, and thus over all the network implements a k-winners take all system, which creates a unique, non-overlapping code for each input. This code serves as the input for CA3.

The DG model module receives conjoined multimodal sensory signals from EC. DG performs pattern separation on this abundance of sensory information to produce sparse output activation, which helps bias different semantic concepts toward unique encodings [140]. Computationally, a series of k-winner-take-all (k-WTA) fuzzy-ART modules constitute the DG module of our model. A WTA module is a competitive network in which a single classification beats out competing classifications to represent the input vector. Effectively, a sparse encoding is created as each of the k WTA modules yields a single output. Regardless of the number of input dimensions, a k element output vector is formed.

DG performs pattern separation as compared to a clustering method. A cluster-

ing method would place all similar input vectors into the same cluster, given $nF2^k$ clusters for k-WTA with nF2 output nodes per WTA submodule. Clustering techniques tend to consolidate patterns. The semi-topological, highly connected, EC output DG input synapse system provides a chance for differences in regions of the input vector to result in classification into a different output vector. Otherwise, regional differences might be outweighed by similarity in the rest of the input when using a traditional clustering technique. Hence we consider this DG architecture to perform pattern seperation.

D.3.3 CA3

Anatomical studies of the hippocampus proper reveal cytoarchitecture which differs radically from that of the cortex [138]. While both CA1 and CA3 contain pyramidal cells like the cortex, existence of extensive recurrent connections in CA3 and the presence of inhibitory and excitatory interneurons have led some investigators to suggest that CA3 may be involved in pattern completion [141] [142]. Building on this idea, CA3 is modeled as a self-organizing map of semantic associations. CA3 uses a modified ART network in which the F2 nodes are laid out on a grid, and updated according to a self organizing map learning algorithm, similar to a Kohonen map [143]. The two dimensional array of modeled CA3 nodes corresponds to the laminar section of hippocampus that is biological CA3.

$$w_j(n+1) = \beta(I \wedge w_j(n)) + (1-\beta)w_j(n) \tag{D.1}$$

The Self Organizing Map ART (SOMART) operates up through the $F2$ layer just as a normal Fuzzy ART. For that matter, SOMART need not use real valued inputs, and therefore need not use Fuzzy ART is particular. SOMART can be adapted from any version of ART. In SOMART, the $F2$ output node which wins selection through through the ART algorithm is updated with fast learning. In fast learning, the ART

learning parameter β is set to unity in Equation D.1. All variables in Equation D.1 are as described when this equation was introduced in section 2.7.1. Neighboring $F2$ nodes are then updates with slow learning, where the speed of learning falls off with distance from the winning node. Learning speed β becomes a function of each $F2$ node's distance from the winning node. In the general case, $F2$ nodes can be laid out on any topology which allows a distance measure.

The operation of SOMART is such that semantic attractor regions are formed in ART template space. The specific mapping for those semantic regions to episodes and phenomena in the environment observed by an embodied SOMART is dependent upon the specific mapping of sensor inputs to SOMART. In general though, so long as sensors which capture an appropriate subset of information from the environment are employed, SOMART semantic regions have representational power for concepts manifest in the observed environment.

The combination of these ART and Kohonen learning techniques is critical. Because the Kohonen algorithm updates a neighborhood of neurons, inputs to CA3 collapse into small attractor basins which represent a particular DG input code. This allows the network to perform pattern completion, as sufficiently similar inputs will collapse into the same attractor-space as previously encoded DG inputs. However, the underlying ART algorithm constrains the Kohonen algorithm to a single, best fit winner, which may not be topologically local to previously encoded attractor basins. Thus the learning algorithm creates islands of activity which respond to similar input sets, but avoids a global topology. These islands of CA3 activation in response to particular environmental stimuli manifest in biological hippocampus as place fields. Place fields are regions of CA3 cells which are repeatably active when a subject is in a certain environmental context, often but not necessarily a physical location [59]. A global topology in the biology would correspond to a map of the subject's known environmental universe within hippocampus. This has logical problems, and is has not been observed.

In the computational hippocampal model architecture, the sparse output pattern from DG serves as input to CA3. Functionally, CA3 assists with episodic binding through auto-association; in our model, this functionality is represented by a self-organizing map (SOM) structure. A standard SOM transforms a given input vector into a distinct topological region without supervision guiding the classification [144]. Incorporating the neighborhood updating capabilities of a SOM within a fuzzy-ART module, a SOMART module has been created to represent CA3. This module is capable of mapping semantically similar inputs to proximate topological regions. In effect, related concepts are clustered together to help associate episodic memories. The activation patterns of these clusters form the input to CA1.

D.3.4 CA1 and Subiculum

CA1 has been implicated in learning relational information for temporal sequences. Since the computational CA3 can only encode momentary conjunctions, a mechanism is needed which can capture sequences of changing relations. Thus, CA1 contains a unit which temporally integrates CA3 outputs using a set of leaky integrators just as occurs in the neocortical model. This provides a temporal gradient of input conjunctions coming from CA3, the oldest bindings will have the weakest signal in the temporal integrator, while the most recent bindings will be most strongly represented. This temporally coded sequence of CA3 activity is used by CA1 to create a topology of temporal sequence.

Once this temporal topology has been established, activity in CA1 is associated with activity in the EC via a LAPART supervised learning paradigm. Local CA1 learning is supervised in that a certain sequence of CA3 activations corresponds with a certain EC activation. CA3 sequence A, where that sequence is translated to an instantaneous representation through the temporal integrator, is bound through learning to EC activation B. LAPART uses two ART modules connected by a lateral activation field, so the activations on each side are generalized via the ART

classification mechanism. Through experience, a connection weight is learned to bind the node that corresponds to each classified CA3 sequence to a node that corresponds to some ERC activation. This mapping of sequences onto the high-level cortical representations closes the hippocampal loop, and allows activations in CA1 to cue top-down recall in the cortex and unspool the temporal representations it has created.

Anatomically, the output of CA3 proceeds to CA1 and then to the subiculum as the major output region of the hippocampus. However, the exact functionality of the subiculum is largely unknown, so the connectivity of CA1 and subiculum have been merged in the computational model. CA1 is involved in forming sequences of relations and connecting these episodic encodings back to the original sensory inputs from EC. This ability to link sequences allows for temporal packaging of episodes. To handle the sequencing of associations, we temporally integrate the inputs passed into the model's conjoined representation of CA1 and subiculum. A laterally primed adaptive resonance theory (LAPART) module represents the conjoined CA1 and subiculum regions. The LAPART module learns to associate CA3 encodings with the original unaltered EC inputs. This allows the LAPART module to complete the hippocampal loop and propagate temporal sequences back to EC and eventually to cortex for long term storage.

D.3.5 Computational Cortical Hippocampal Process

Figures D.2 through D.11 illustrate the process of neural activations through the hippocampal model. Examples are shown of behavior through sensory cortex, endorhinal cortex, detate gyrus, CA3, and CA1. Later, in section D.4, results from the experimental implementation of this model are presented.

In Figure D.2, when presented with a novel input pattern, an ART module creates a representative category. The temporal integrator connected to the ART module raises the activation of the corresponding category to full value, encoding it as the

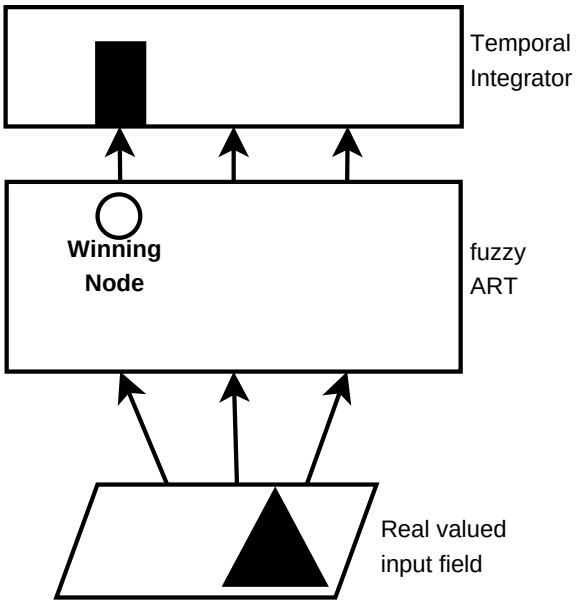

Figure D.2: Initial input into cortical model.

most recent F2 activation.

Let the same input pattern be presented, once again, as illustrated in Figure D.3. The ART module uses the same category encoding and the corresponding temporal integrator activation remains at full value.

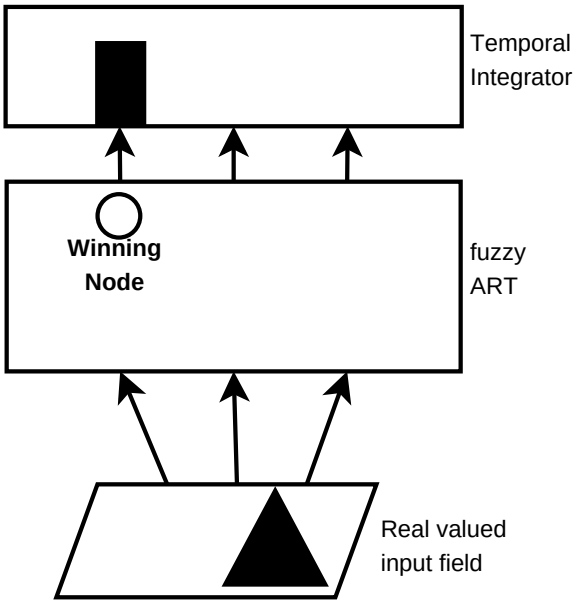

Figure D.3: Second, identical, input into cortical model

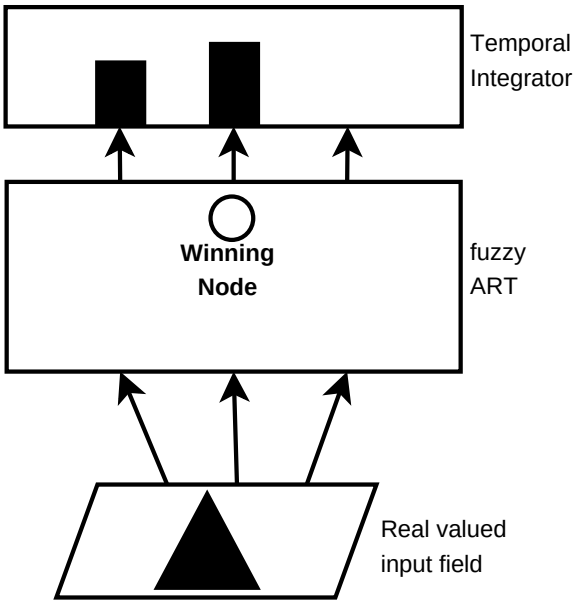

Figure D.4: Third input into cortical model, abstracted to different cortical class than first two inputs.

Next, in Figure D.4, a new input pattern is presented. This input pattern is sufficiently different from previous pattern that vigilance is not met, i.e. the input vector is too far from the closest template vector. The ART module forms a new category and primes the corresponding temporal integrator value. The temporal integrator decays the activation value of the previous category and the current category's node is active at maximum value.

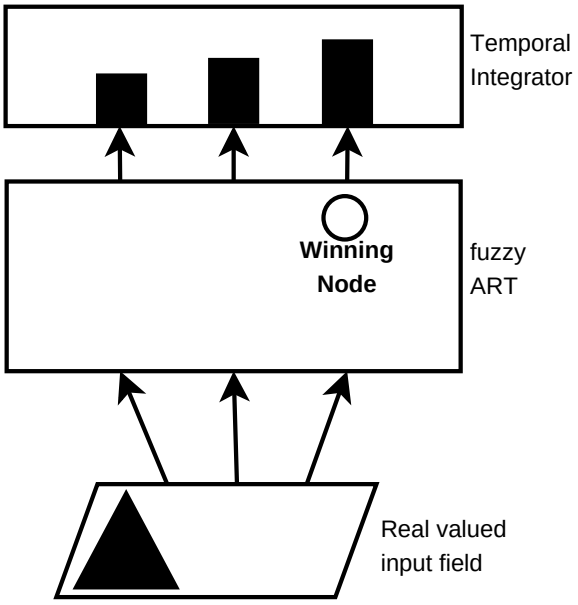

Figure D.5: Fourth input, third distinct input. A third distinct $F2$ output node is active.

With the presentation of a third distinct input pattern in Figurefig:HCprocessD, a corresponding third category is formed which primes the respective temporal integrator value. The temporal integrator decays the activations of previously observed categories creating a real-valued gradient vector of temporally integrated categories.

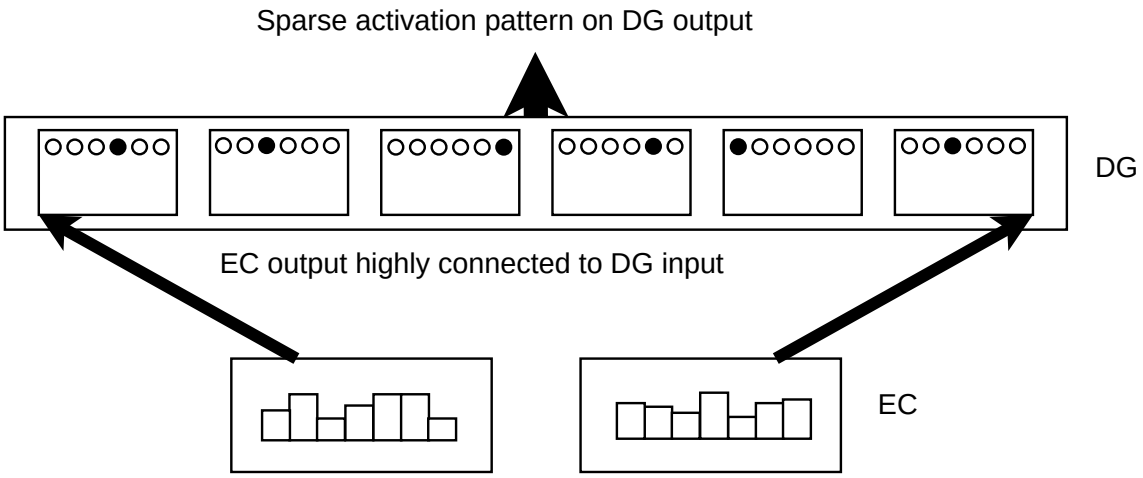

Figure D.6: Consolidated activations from multiple areas of sensory cortex converge upon endorhinal cortex and are linked through a highly connected synaptic network to dentate gyrus. The high connectivity between EC and DG, in combination with k-winner-take-all behavior for DG output neurons is the modeled mechanism for sparse activation of DG neurons.

Figure D.6 illustrates converged sensory cortex activations, which represent observed semantic concepts, being encoded as a sparse output activation by DG. Where we recall that DG is comprised of a number of individual classifiers which a highly connected to EC outputs. Through cortex, layers of sequential ART modules and temporal integrators allow for the creation of categories of categories, which further allow for the formation of semantic concepts in the EC as the convergence of sensor modality input patterns. Multi-modality semantic concepts from EC are passed as inputs into the DG which yields a sparse pattern of output activation.

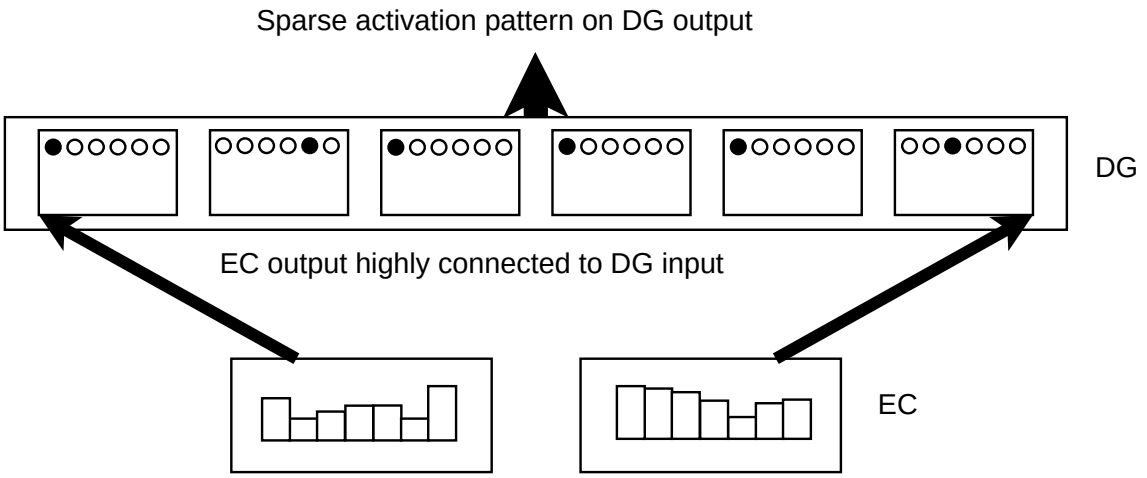

Figure D.7: Same model elements as the previous figure. The figure illustrates unique DG activation as a result of unique DG activation.

As seen in Figure D.7, unique DG inputs create distinctly different sparse activation patterns. The hypothesized function of DG is a form of pattern separation, as supported in the neuroscience literature [139] [145]. Highly converged semantic encodings are input and sparse lower dimensionality encodings are output.

The separated sparse activations serve as inputs to the highly recurrent CA3 hippocampal region which processes these inputs performing an auto-associative mapping as represented in Figure D.8. Recurrence is observed in biological DG [146], and the form in which we model recurrence is in both the function of ART and the local activation reinforcement of the self-organizing map portion of SOMART.

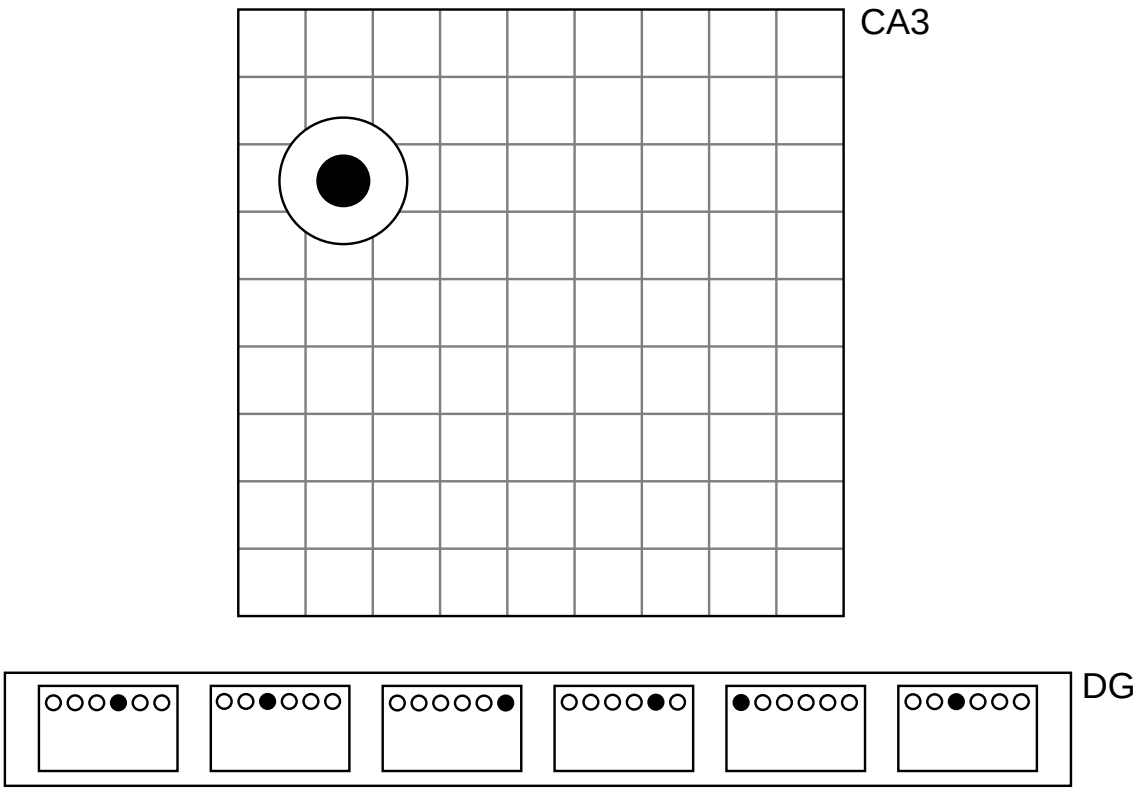

Figure D.8: Illustrates one possible DG activation pattern and a corresponding field of activation that might form in the SOMART modeling CA3.

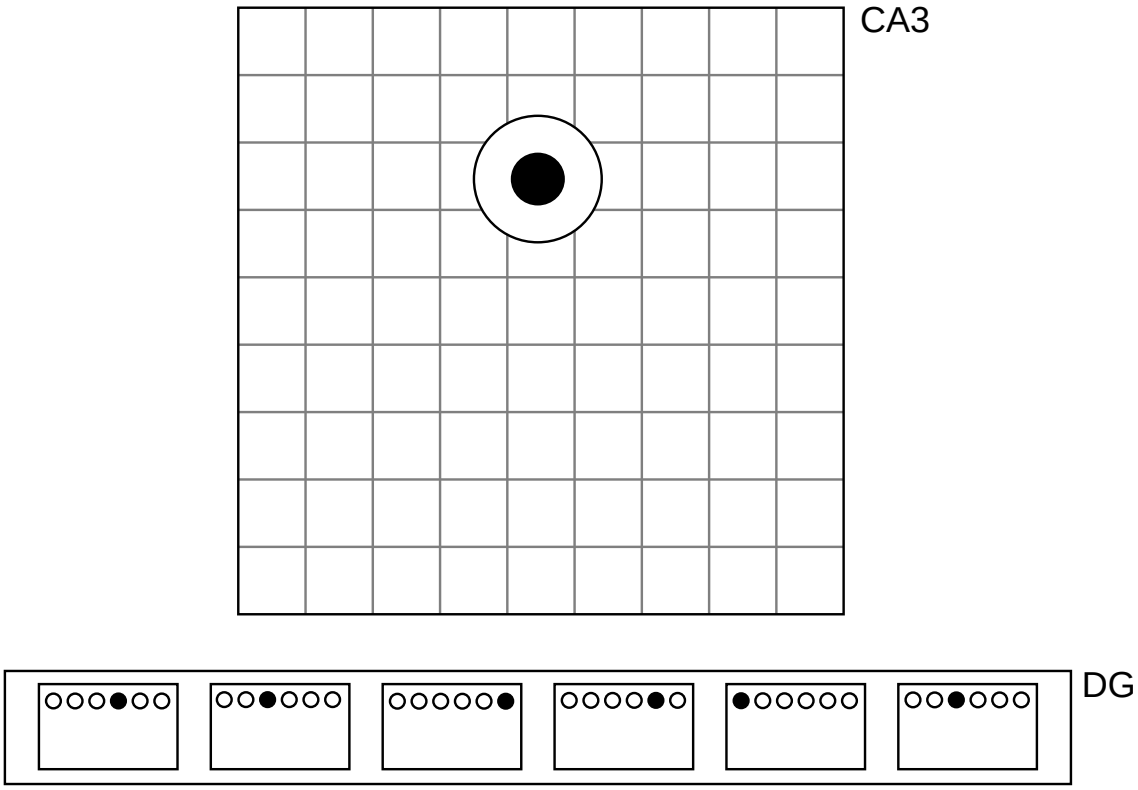

Figure D.9: Second, different, input into CA3. Different DG activation pattern results is different characteristic activation field in CA3 output array.

A second, different input propagates to CA3 in Figure D.9. Unrelated sparse activations are observed to be mapped to distinctly different CA3 regions, and related patterns are mapped in close proximity to each other facilitating association of related concepts. This mapping creates a topology across the CA3 output space. The particular topology is an explicit function of the neuromorphic connectivity of the cortical and hippocampal model components. Implicitly, the topology is a function of the semantic content of sensory observations.

In Figure D.10, CA3 auto-associations are integrated temporally within CA1 and create an input vector for the A side of a LAPART. This allows relational mappings to form between associated temporal semantic episodes and EC semantic concepts.

Figure D.11 illustrates that localized regions (islands) of activation are formed

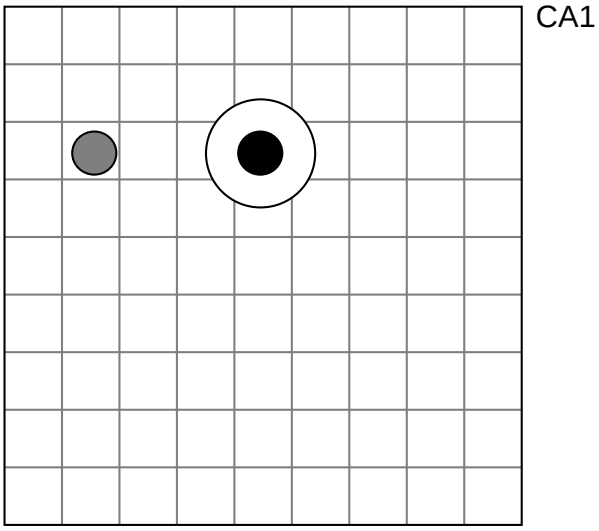

Figure D.10: The input nodes of CA1 temporally integrate the output activations from CA3. This results in CA1 associating temporal sequences of CA3 output fields to instantaneous EC activation patterns. The EC activation patterns are incident upon the B side of the LAPART which models CA1, while the temporally integrated CA3 outputs are incident upon the A side.

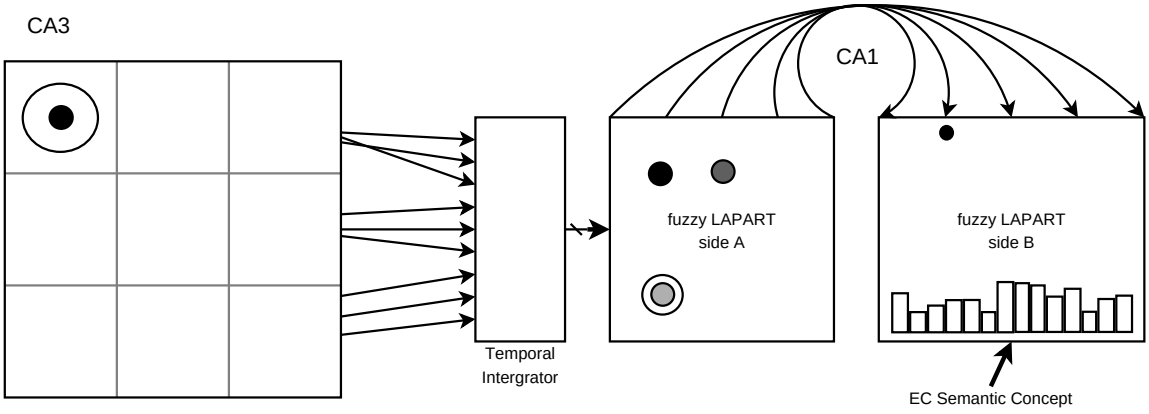

Figure D.11: Temporal encoding in CA1.

temporally as prior regions of activity decay away. This decay process contributes to distinct episode encoding, along with the abstraction of sensory inputs that results in changing ART $F2$ output nodes. CA1 can perform prediction in the form of temporal pattern completion based on prior experience.

In the computational hippocampus architecture, when an episodic memory is re-

called, the hippocampus activates one of the categories in CA1, and it begins the top-down recall of the episodic trace which it encodes. In that recall, the active category in the CA1 ART unit encodes a temporal sequence of conjunctive associations which were formed in CA3 during the episodic memory formation. During recall of a specific conjunctive association in CA3, each stream (ventral and dorsal) that makes up part of the conjunctive association is activated in top-down fashion. The top-down activation of each stream is similar, thus the top-down description of a single stream will be given in the interest of brevity. In the top-down activation of a cortical stream (either focus or context), the top-level ART category is activated (in the EC or DG). This top-level category consists of a concatenation of ART categories from each column in the cortical stream. Top-down recall of this concatenation of categories consists of a simultaneous recall of each column starting at the ART unit which is at the top of each column. It is this ART unit which provides its category for the concatenation during the formation of the episodic memory. The top-down recall for each column is similar, thus the description of a single column will be given in the interest of brevity. During recall of a cortical column, the category in the top-most ART unit is activated in top-down fashion. This category contains a temporal sequence of categories from the next lower ART unit as integrated through the integration unit between them. The recall continues from ART unit to ART unit downward through the connecting temporal integration units until the bottom ART unit is reached. In top-down activation, each temporal integration unit contains a temporal sequence of ART category activations which were fed to it as input during episodic memory formation. During recall of the temporal sequence of ART categories in a temporal integration unit, each category in the sequence is re-activated in top-down fashion in the same temporal order as was originally experienced and encoded in the episodic memory trace. When the recall reaches the lowest level ART unit, it is ready for "replay." During replay, the memory is re-activated in forward or bottom-up fashion in the same temporal sequence it was originally experienced.

In this system, temporal information is stored in the activation potentials of temporal nodes. Local semantic information is stored in the synaptic weights of the ART modules. Long-term, memory can occur through Hebbian-like adaptation of synaptic connection weights between local cortical areas. In our model, a local cortical area at a given level is comprised of a collection of nodes that all influence the activation of the same ART output node.

D.4 Experimental Results

D.4.1 Associating Object/Scenes Pairs

For this assessment of the cortical hippocampal model, results from the computational model are compared to published human study results of Hannula et al. [147]. The Hannula study, as illustrated in D.12, presented human subjects with a series of face-scene pairs in a study block. Eye movement was then tracked for sets of three faces, with a background scene, presented in a test block. The set of three faces can be from one of three categories: match, re-pair, or novel. The match face sets contain three known (previously seen) faces, one of which is correctly paired with the background scene. The re-pair face sets contain three known faces, but none of them are correctly paired with the background scene. The novel face sets contain three unknown faces. This task is an exercise in episodic memory for associating people and places.

Published results show that normal subjects will spend a larger proportion of viewing time directed to a face that correctly matched the background. Subjects with hippocampal damage did not exhibit this proportional increase in dwell time on the matching face. This result indicates that hippocampus is required for the recognition of previously observed episodes.

The goal of this assessment is to show evidence that our model exhibits some of the same function as biological brain with regard to scene/object pair association.

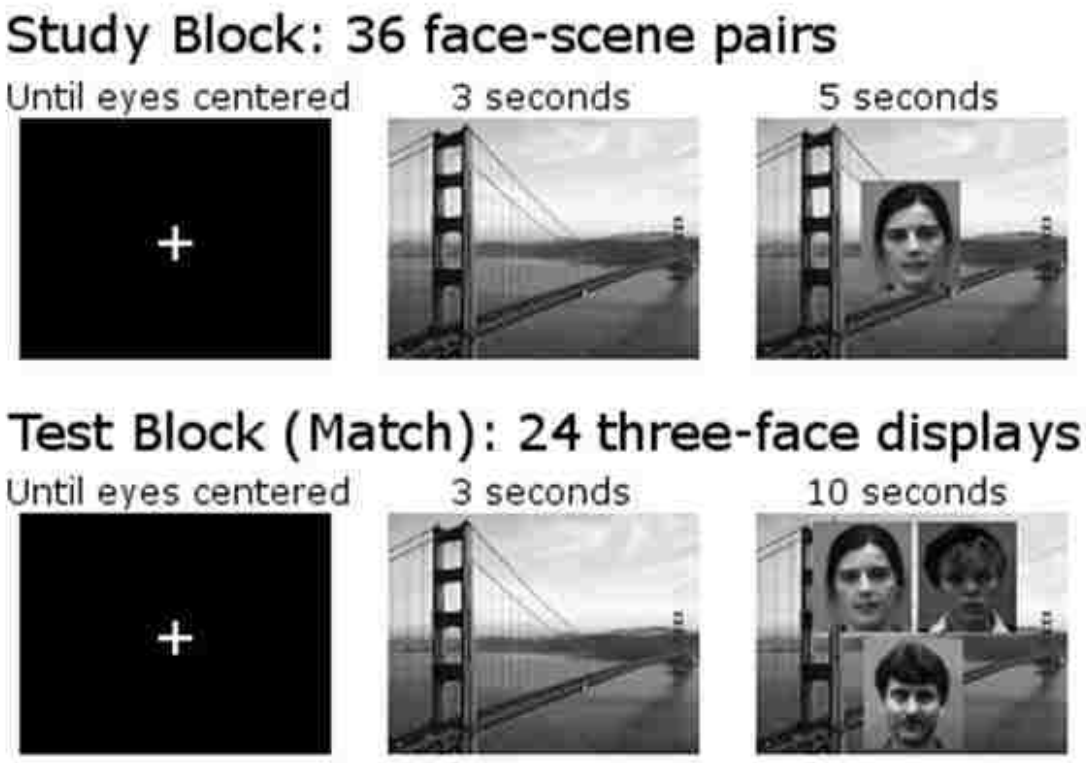

Figure D.12: Visual input scheme of Hannula study.

In the interest of correlating behavior from our model to human behavior, we create a mapping of the human experimental setup to an experiment that we can run, in simulation, on our model. To mitigate visual processing effects, we map the face-scene focus-context images to simple geometric images (initially using squares and triangles, then going to orthogonal lines and dashed lines). Our input images were ten pixels by ten pixels.

We present arrangements of our focus and context images that correspond with the study and test image presentations of the original experiment. The study presentation sequence of the original experiment is fixation, scene, face. As the computational model lacks a mechanism for separating focus and context information in the visual field, we must simulate that separation by presenting separate images

to the focus and surround modality inputs. The presentation sequence for the focus modality is fixation, scene, face. The presentation sequence for the context modality is fixation, scene, scene. These sequences reflect the fact that the scene image is a focus image during the second element of the original study presentation because the scene is the only image on the screen. The context modality only ever sees the scene because even when the face is present in the original study presentation, the scene still forms the background of the image. Part of the original experimental setup is that visual dwell time on an image is a measure of recognition of that image. As the model is lacking eyes, an alternate measure of recognition must be developed. The simulated measure of recognition is equivalent to directly probing neural activation in a human brain. Modeled neural activation can be evaluated by observing the ART classifier module output in the cortex model and the grid node outputs in the hippocampus model. Model recognition scores are computed by summing contributions from each cortical classification module, and the hippocampus. The cortical classification modules can each contribute one point, and the hippocampus can contribute a point. The first ART module in the cortical focus modality contributed a point if it identified an existing template (i.e. it had previously learned a generalization) for the current input. As inputs are presented in sequences of three (fixation, scene, face), the first ART module will make its contribution based on the last element of the sequence. The second ART module in each cortical modality is located after a temporal integrator, and so it will score familiarity based on the whole sequence.

Another biological brain mechanism that the computational model lacks is the ability to concentrate on different portions of an image. As such, that ability is simulated for the purpose of the test images. Instead of a single sequence with the last image containing three faces, three sequences are presented with the last image each containing one face. This way, the model does not need to consider three sub-images as the human subjects do when looking at the single test image of three faces.

D.4.1.1 Empirical Results

Figure D.13 shows an example of the experimental results. This example reflects two test sequences from the Hannula experiment. As this example is from the match category of faces, all three faces are known. One face should have an episodic memory associating it with the tested scene.

Presentation timestep	Focus-Input Index	Context-Input Index	Total Familiarity Score
1	1	1	5
2	2	1	4
3	3	1	4
4	2	2	5
5	5	2	4
6	6	2	4

Figure D.13: Results of Hannula experimental comparison.

The intact model results in higher familiarity scores when previously studied matching focus-context (face-scene) image pairs are presented, as opposed to pairs that were not studied together. This behavior correlates to the eye dwell time of human subjects in the Hannula study. If the hippocampus section of the model is lesioned, familiarity scores are the same between matching and non-matching image pairs. This behavior also correlates with human subjects, where subjects with hippocampal damage do not preferentially dwell on particular faces during the matching and non-matching face-scene pairs. Further results show that there is no difference in familiarity scores between different focus images in the re-pair and novel tasks, with either intact or lesioned models. These results correlate with the Hannula data where hippocampally damaged and normal subjects both view faces in the re-pair task with no preference.

D.4.2 Co-occurrence of Shared Scenes with Novel Objects

For the third assessment we compared the model to a study performed by Preston et al. [148]. In the Preston study, human subjects were trained on black and white photographs of face-house pairs and face-face pairs in three sets. The first set consists of pairs of faces and houses. The second training set introduces new faces paired with the same set of houses shown in the first set. And finally, the third training set consisted of face-face pairs which were previously unseen. During the testing phase of the Preston study, subjects performed forced-choice judgment tasks. Two of the tasks presented either a face or a house and required the subject select the corresponding house or face to complete the pair. The other two tasks focused on face-face pairs. One task was simply a test of the learned face-face pairs, whereas the other task tested subject's ability to recall related face-face pairs which shared a common house but which were never explicitly seen together. Similarly, we trained our model using face-house pairs such that a face is processed by the ventral stream and a house is processed by the dorsal stream. An example of the input presented to the model may be seen in the left half of Figure D.14. See Appendix B for the full training sequence. Our model lacks the ability to perform the forced-choice judgment task. So rather we first trained the model on face-house pairs including faces with a common house. Then, we turned off learning in the model so that no new concepts could be formed, but rather only existent concepts could be used. We then presented the model an ambiguous partial input cue by inputting a blank image to the ventral stream and one of the houses previously seen during training to the dorsal stream. This partial cue presentation may be seen on the right half of Figure 15. Rather than selecting between possible choices as in the forced-choice judgment task, we get our model to reconstruct the image that it has stored in memory associated with the house. The resultant image is not an exact copy of the original input, but rather is an amalgamation of categorical representations distributed throughout the hierarchy of TIART modules comprising sensory cortex. A sample recalled face may be seen

at the bottom right of Figure D.14.

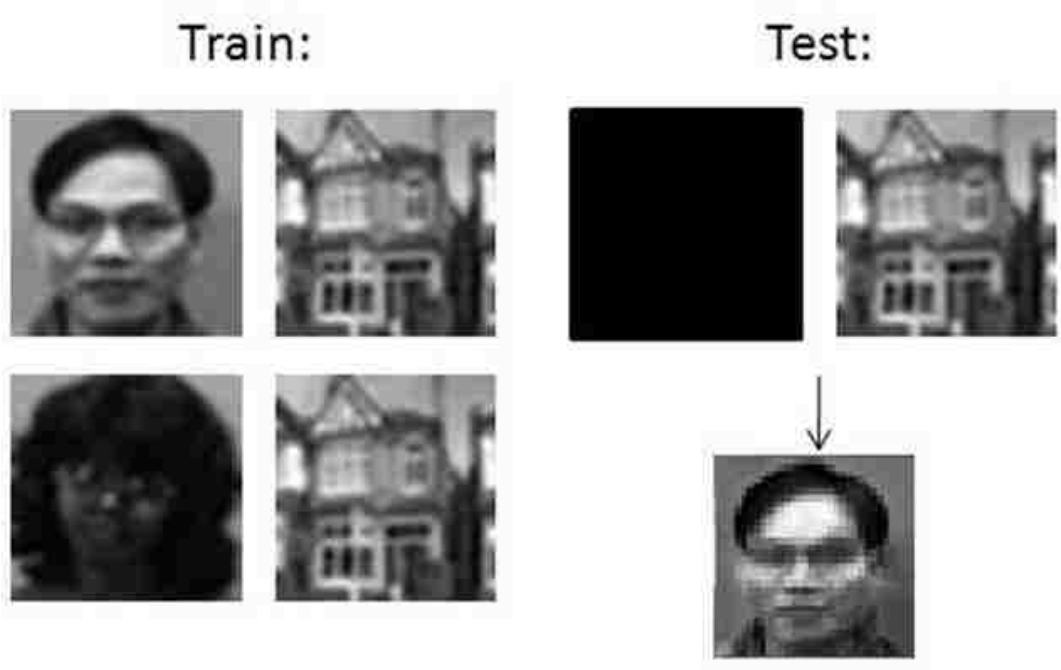

Figure D.14: Visual input scheme of Preston study.

D.4.2.1 Qualitative Preston Comparison Results

In addition to the example shown in Figure D.14, the model was successfully able to recall correct corresponding faces for each of the houses shown during testing. All of the recalled faces clearly resembled a particular face shown during the training phase; however, each was subject to slight distortions yielding an imperfect recall. The ability to associate related face-face pairs can be observed qualitatively within the model by noting the activation regions within CA3. Portrayed in the upper left region of the graphical user interface (GUI), shown in Figure D.15, informally one can observe whether or not the same CA3 activations are employed to encode the association of cohabitation. The visual input presented to the model can be seen in the lower left of the GUI. As displayed in Figure D.15, when presented with face

A and house A, a distinct region of CA3 is activated. Likewise, as shown in Figure D.16, when presented with a different face B also paired with house A, an overlapping region of CA3 is indeed active indicative of the shared encoding between the related face-face pairing. On the other hand, as shown by Figure D.17, when presented a distinctly different face C and a different house B an entirely different CA3 region is utilized for the encoding.

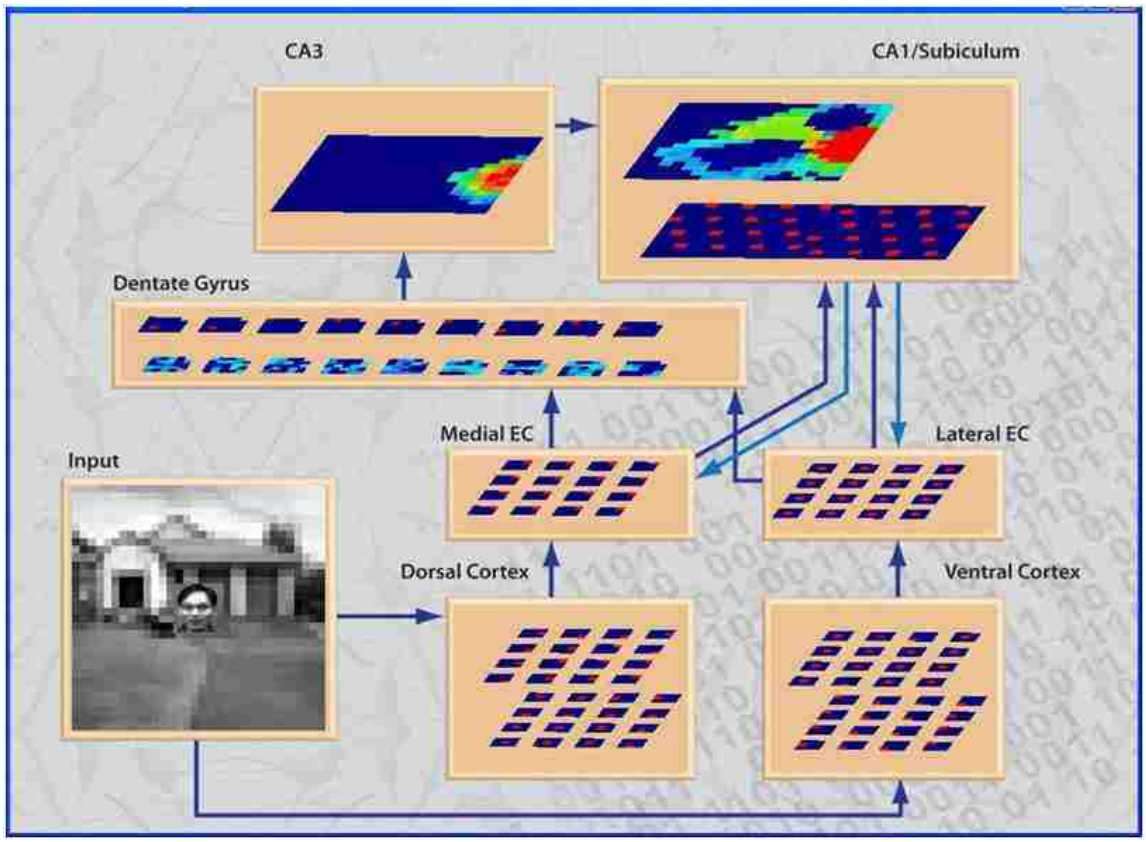

Figure D.15: Person 'A' with house 'A'

D.4.2.2 Quantitative Comparison Results

In the absence of a fully embodied model with output modality to articulate the envisioned associations beyond cued recall, a mathematics technique from informa-

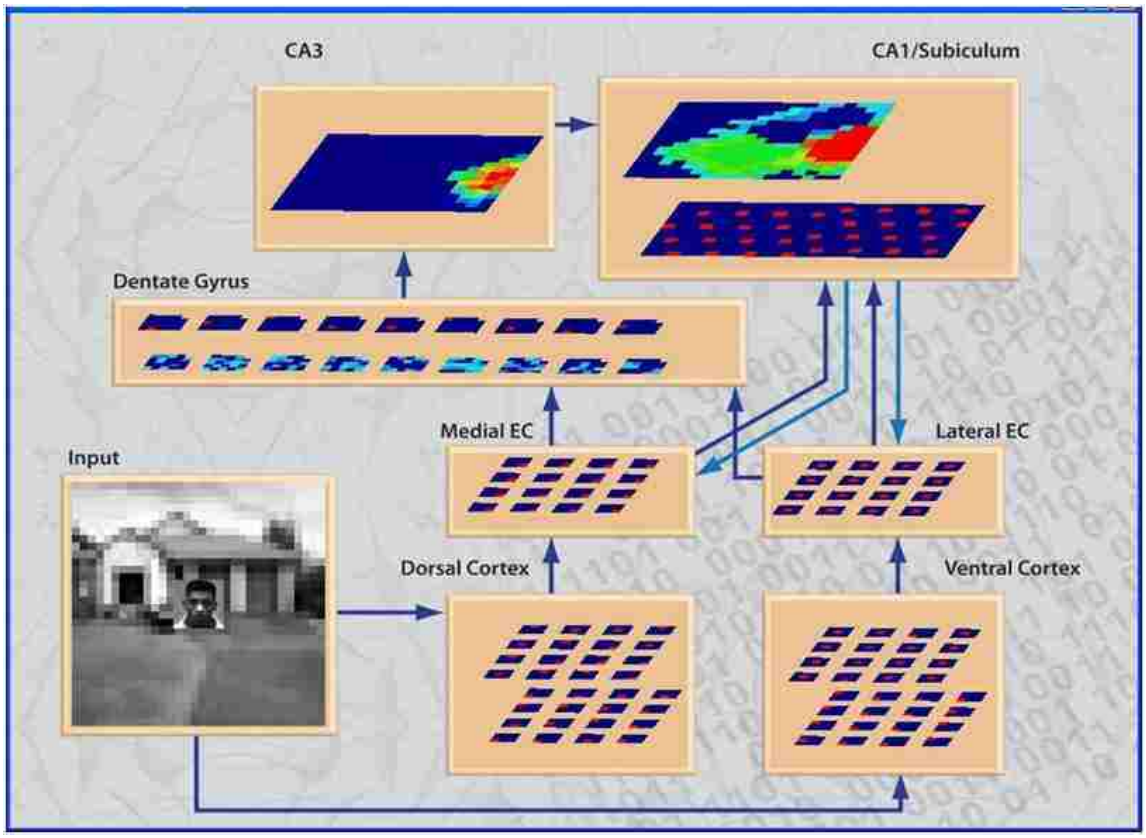

Figure D.16: Person 'B' with house 'A'

tion theory has been applied to quantify the relationship between semantic concepts within the architectural implementation of CA3. Information theory allows for a quantitative evaluation of the information content independent of the particular computational implementation or the underlying neuroanatomical processes modeled. More specifically, within the context of information theory, mutual information is a measure of the dependence between two random variables [149], and is computed by the double summation given in Equation D.2.

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) log(\frac{p(x,y)}{p(x)p(y)} \tag{D.2}$$

$I(X;Y)$ is the mutual information between random variable X and random variable Y. The random variables are states of the output neurons of CA3. x and y

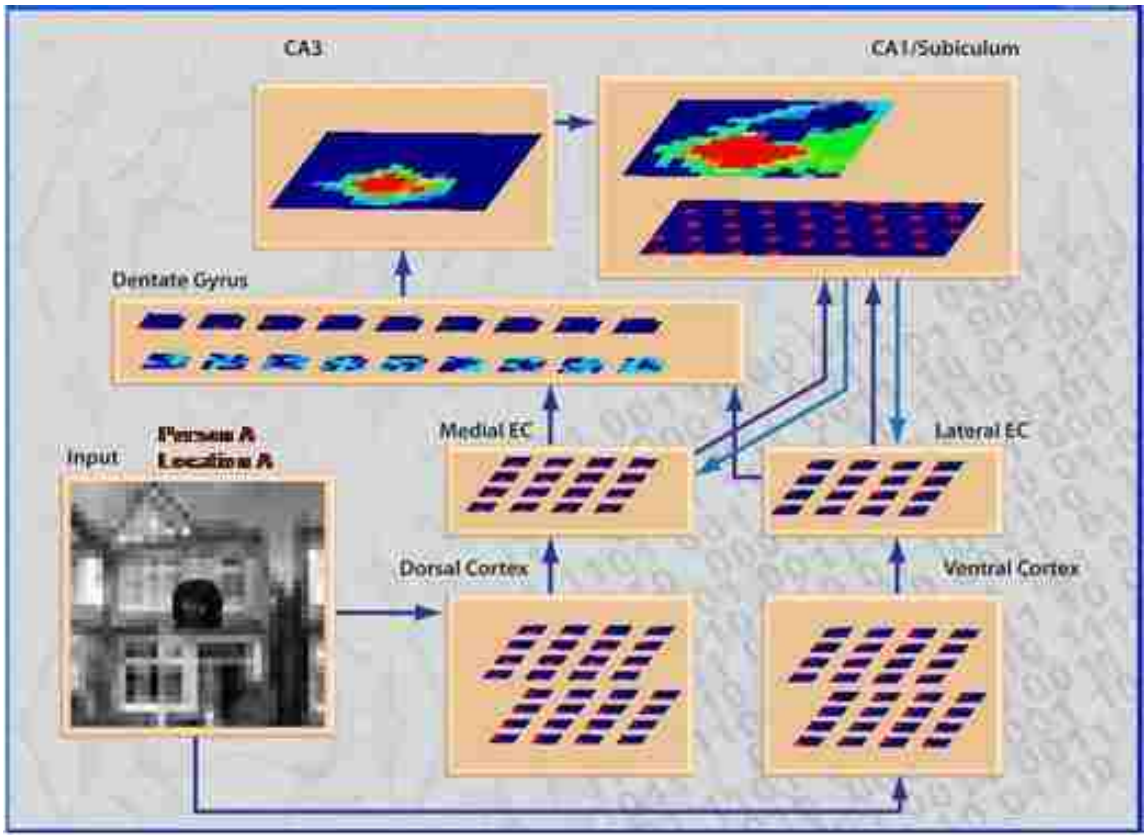

Figure D.17: Person 'C' with house 'B'

are neural nodes in the respective random variables, e.g. a CA3 instantiation with a ten by ten output node array would have one hundred x and one hundred y variables to sum over in the double summation. Probabilities are calculated as the node activation value, using the convention that a node activation level represents the likelihood of an action potential event. $p(x)$ is the activation value of node x. $p(y)$ is the activation value of node y. $p(x, y)$ is the joint probability of x and y, which is calculated as the product of the activation of x and the activation of y.

Treating a conjoined face-house pair semantic concept as a random variable, the various CA3 encoding regions represent the alphabet of possible states the random variable may take on. In other words, a single random variable X represents the combined concept of a particular face and a specific house. Any specific pattern of

activation within CA3 may be used to represent the random variable, and thus the entire CA3 grid is the set of all possible values the random variable may express. From this perspective, mutual information may then be used to quantitatively evaluate whether or not the architecture recognizes and auto associates inferred relationships.

In comparison to the Preston study, this technique allows us to evaluate whether or not our model is capable of forming an association between unseen related face-face pairs. A single face is only part of a random variable, and so for two different faces to share a relationship they must have a common context. The left column of Figure D.18 lists the mutual information for the related face-face pairs our model was trained on. For instance, the first entry is the mutual information for two random variables A and E. A different face is represented by A than that of E; however both random variables share the same house. The right column on the other hand presents an averaged mutual information value of all the unrelated faces in reference to a particular face. As an example, the face represented by A is only related to the face represented by E. All other random variables (in this case B, C, D, F, G, and H) represent unrelated faces. Therefore, in column 2 of Figure D.18, we represent the average mutual information values for non-matching (i.e. non-auto-associated) faces.

Furthermore, we have tested the capabilities of our model on even more complex associations than those in the Preston experiment, to demonstrate the flexibility available in forming novel arbitrary associations. As shown in Figure D.19, we have tested our model using a vehicle context in addition to houses. In addition to contextual relationship, a more advanced partially overlapping association occurs in this more advanced example.

The motivation behind the Preston et al. study was to investigate the role of the human hippocampus in the novel expression of declarative memories [148]. Comparable performance by our model on an equivalent test demonstrates a functional appropriation to the modules comprising our architecture.

Mutual Information of Related Face-Face Pairs	Average Mutual Information of Unrelated Pairs
I(A;E) = 0.3657	I(A;~E) = 0.0254
I(B;F) = 0.3628	I(B;~F) = 0.0303
I(C;G) = 0.3303	I(C;~G) = 0.0294
I(D;H) = 0.3570	I(D;~H) = 0.0322
I(E;A) = 0.3657	I(E;~A) = 0.0247
I(F;B) = 0.3628	I(F;~B) = 0.0293
I(G;C) = 0.3303	I(G;~C) = 0.0296
I(H;D) = 0.3507	I(H;~D) = 0.0323

Note: ~ denotes negation

Figure D.18: Model comparison to Preston study.

In the Preston study, human performance was near perfect for the learned face-house pairs [148]. Likewise, the computational model was successfully able to recall a correct face for each house presented such that the recalled image incurred only slight distortion. Due to algorithmic limitations, the present version of the model can only recall one of the faces associated with a given house. However, this could be corrected by allowing the model to retrieve all association pairs rather than only the single best match.

The Preston study observed increased hippocampal activation during fMRI scans when subjects were tested on related face-face pairs compared with learned face-face pairs [148]. This observation demonstrates the important role of the hippocampus in relational tasks. While we are not concerned with relational tasks in isolation, we are concerned with high fidelity modeling of hippocampus. As such, empirical evidence of correlation between our model and biological behavior is useful in any

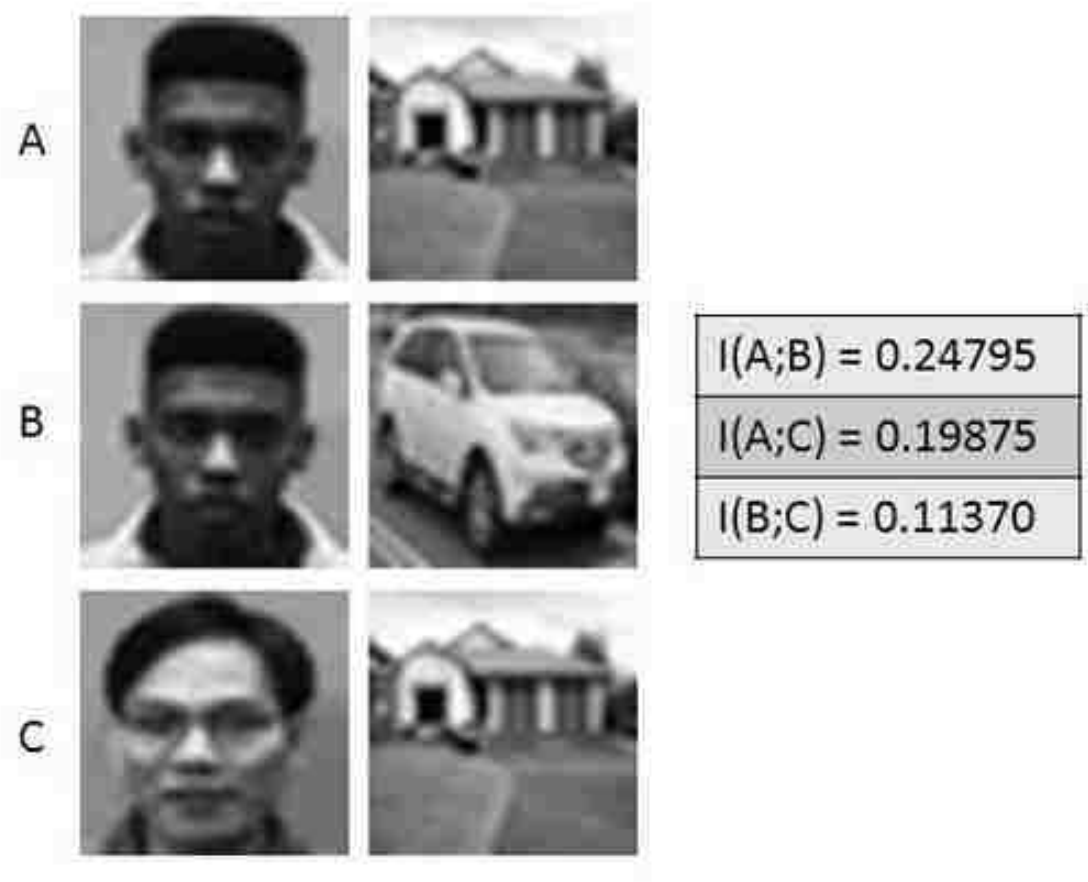

Figure D.19: Faces paired with different contexts.

form as validation of the computational methods.

Beyond simply leveraging the hippocampus to form associations that enable desired function within the model, mutual information has quantifiably shown the model is capable of forming associations between novel concepts. In our mutual information measure, we approximate the joint probability distributions for two semantic concepts. This approximation is calculated by computing the normalized fuzzy conjunction of the respective CA3 activations when the concepts are processed by the model individually. As can be seen by the mutual information approximation values given in Figure 20, the related face-face pairs have a significantly larger CA3 mutual information measure than that of unrelated pairs.

Furthermore, by incorporating vehicles as an additional context, we are able to demonstrate that our model is capable of processing a variety of concepts, as is true for humans, and is not only capable of processing houses. This more complex association additionally demonstrates the ability to associate multiple contexts with a single focus in addition to associating multiple foci with a single context. For example, as illustrated in Figure D.19, the same person represented in concepts A and B is associated with a house in one concept and a vehicle in the next. A second person is additionally associated with the same house as shown in concept C. While both people cohabitate the same house, only the first person is associated with the vehicle. The ability to differentiate between these overlapping associations is evident by the mutual information measures. Both the mutual information value associating the first person with his house and vehicle, as well as the mutual information value associating the two people cohabitating the same house are considerably larger than the relationship between the second person and the vehicle. Comparing human CA3 activation during a co-occurrence task [150] to the CA3 activation of the computational model during a similar co-occurrence task yielded similar results. This is shown in Figure D.20.

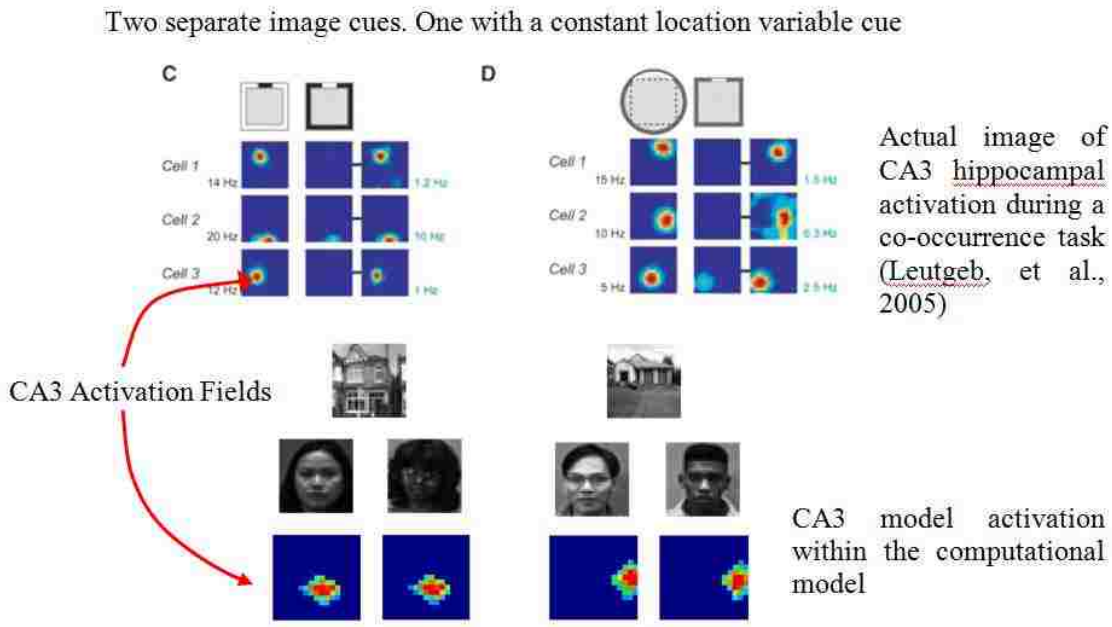

Figure D.20: Experimental and model image of CA3 hippocampal activation during a co-occurrence task.

Appendix E

LimitsART Experiment

This appendix presents additional detail to the modification of Fuzzy ART which enables representation of categorical structure that otherwise would not be possible. The categorical structures which cannot be represented by unmodified ART are limits and colimits. First, the inability of ART to represent categorical limits is discussed. Then, an ART modification is described that allows categorical limit structures to be represented within an ART system. This modified ART is labeled LimitsART. To finish this section on the development of LimitsART, experimental results are presented to show improved performance of LimitsART versus general ART. These experimental results motivate the incorporation of LimitsART into the embodied architecture.

E.0.3 ART and Colimits

This subsection discusses categorical limit structures, why they are useful, and the inability of unmodified ART to represent then. Categorical limits are a critical element of structure in category theory, and it is hypothesized that the embodied architecture will perform better if it has the capacity to represent transforms of categorical limits from the **Concept** category.

The following ART modification was originally described and tested with stack

coded inputs on binary ART1 [115]. As fuzzy ART is used in the neural architecture being developed, a real valued description is given here. Binary ART 1 has been shown to be equivalent to fuzzy ART [66] in classification template formation, therefor an initial understanding is that a valid transformation exists from the pre-existing binary stack coded description to a real valued description.

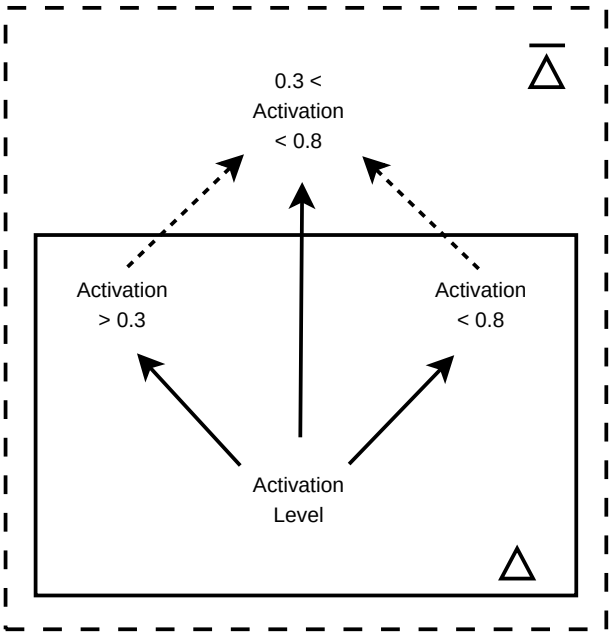

Figure E.1: A stimulus concept colimit with base diagram Δ and defining diagram $\bar{\Delta}$. The apical object $0.3 < Activation < 0.8$ could also be represented as a stimulus interval object $StimInt(0.3, 0.8)$.

Let r be an value with $1 \leq r \leq N$. For fuzzy ART, N is at most the maximum encodable float value in the computational neural implementation. Another popular choice for N would be 1 in a normalized fuzzy ART implementation where all input vector elements are between zero and one. For binary ART1, N would correspond to the number of positive stack nodes in a stack encoding of r. For each pair of values (q, q') with $-1 \leq q$ and $q' = q + r$, we can form a colimit as illustrated in Figure E.1 for the specific values $(0.3, 0.8)$ and in Figure E.2 for the unspecified values (q, q').

The colimit apical object $StimInt(q, q']$ is a concept that combines $StimLowerBound(q)$

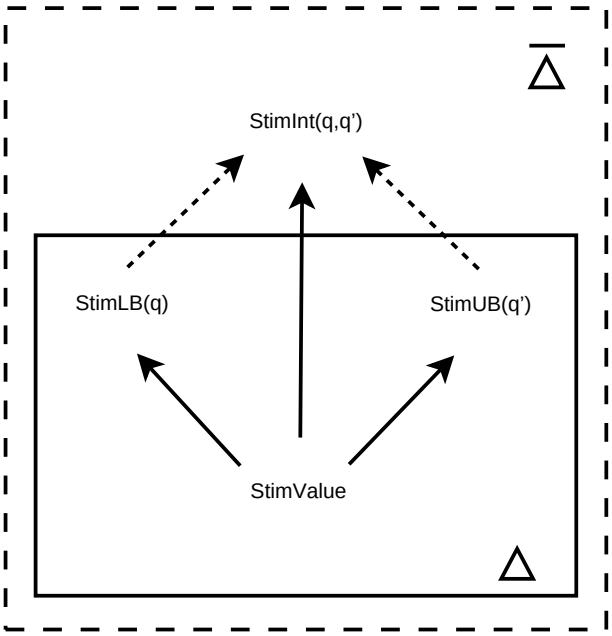

Figure E.2: A stimulus concept colimit with base diagram Δ and defining diagram $\bar{\Delta}$. The apical object $q < Activation < q'$ could also be represented as a stimulus interval object $StimInt(q, q')$.

with $StimUpperBound_{q'}$ at their common sub-concept $StimValue$. Note that a structural relationship is being described between these concepts. $StimValue$ is more general than $StimLowerBound$ or $StimUpperBound$ in that $Value$ has not been tied down as either an upper bound or a lower bound. A yet more general concept might be $Value$, where is has not been specified that the value is part of a stimulus condition concept. These relationships are examples of the utility of categorical limit and colimit representation within a neural architecture. Without the categorical limit and colimit structures, there is a fundamental inability to represent these concepts, and therefore in the framework of the CNST there is no mapping from the **Concept** category to the **Neural** category.

The LimitsART experiment tests the hypothesis that LimitsART can yield improved performance over ART on the type of problem it is meant to address. The experiment will produce evidence as to whether control of hyperbox side length is a reliable way to generate hyperboxes for clusters that are naturally separated into

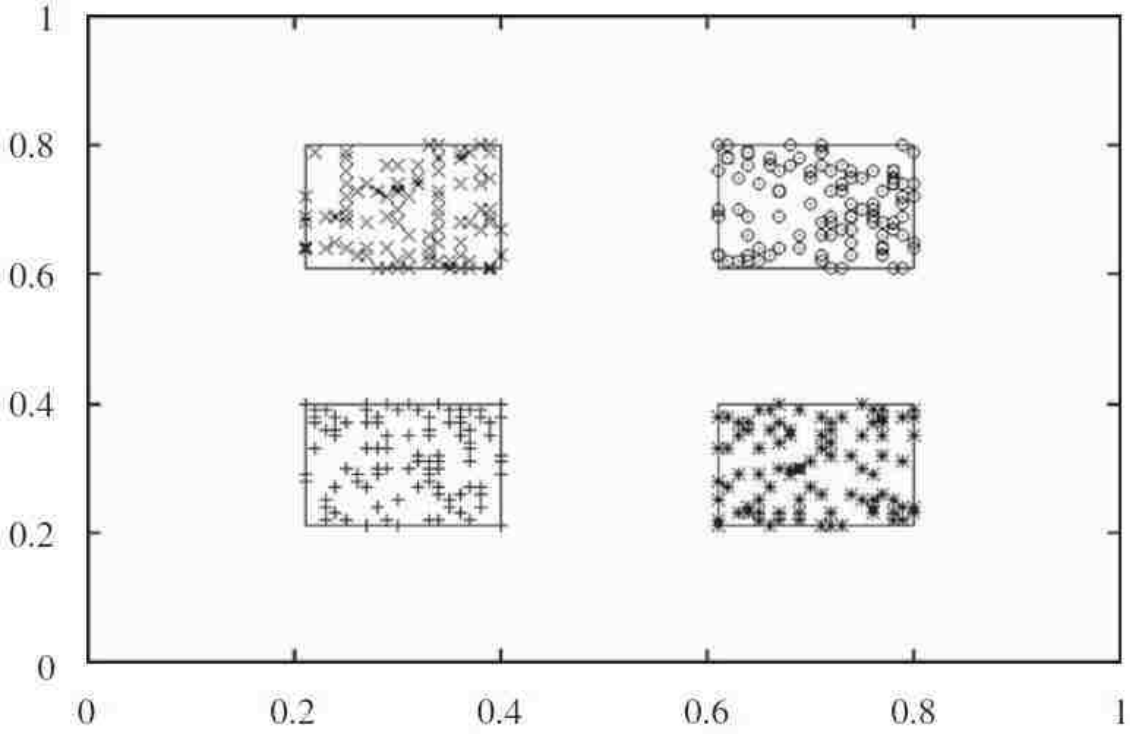

Figure E.3: Cluster hyperboxes (where $t = 0.0$) with $\rho = 0.8$.

hyperbox regions with fixed side lengths in each dimension. This experiment is not meant to be a comprehensive characterization of LimitsART, but rather to supply input as to whether the above design for LimitsART accomplishes its objective of improving performance over unaugmented ART.

To make this as simple and unequivocal a test as possible, the experiment data are separated into clusters which are circumscribed by 2-dimensional square hyperbox regions, corresponding to the use of a single threshold t across all dimensions. The minimum separation between clusters is equal to the side length, and this separation occurs in both dimensions (see Figure E.3 and Figure E.4). Testing the hypothesis serves two purposes: first, it illustrates what in general is to be gained by controlling hyperbox size. Second, it provides a clear exercise with the minimal expected difficulty in obtaining this gain: forming the appropriate hyperboxes for clusters as well separated as described must be a minimum requirement for improvement.

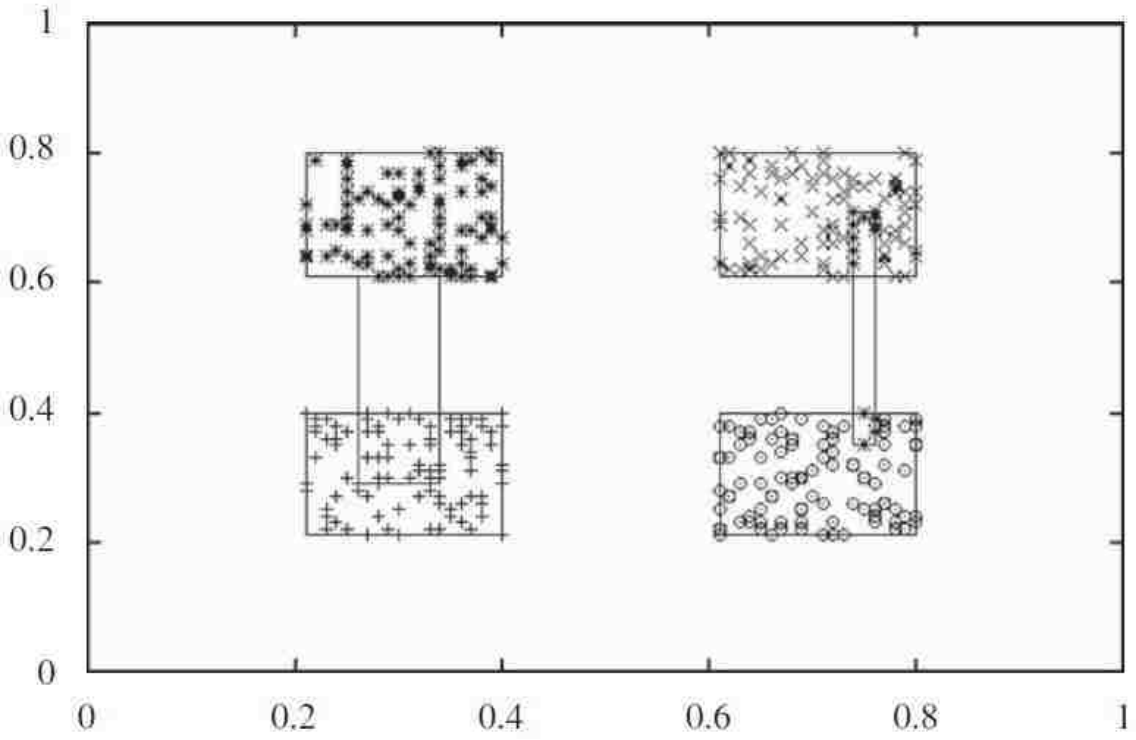

Figure E.4: Cluster hyperboxes (where $t = 0.8$) with $\rho = 0.8$.

E.1 Experimental Procedure

The neural network data were constructed to represent points in 2-dimensional Euclidean space, with a natural spacing into four clusters separated as suggested. Each point was represented by a real-valued pattern of $n = 2$ components, corresponding to two $F0$ stack interval segments representing two hyperbox dimensions. A square rectangular region was defined and 400 data points generated within this region as follows. First, four square rectangular subregions were defined within the larger region to serve as a base for defining four well-separated clusters. Each cluster was contained within one of the subregions. The larger region had min/max coordinates 0.0/1.0 in both dimensions. The four smaller regions were defined to have min/max coordinates at 0.2/0.4, 0.6/0.8 in the two dimensions, resulting in four square rectangular subregions, one for each cluster (see Figure E.3 and Figure E.4). The resulting

separation distance in each dimension between the four cluster regions was 0.2 in at least one coordinate, equal to the length of each side of each subregion. Thus, the minimum spacing between clusters was as great as the width of a cluster in either of the two dimensions. The points in each of the four clusters were generated by applying a pseudorandom number generator to obtain 100 points according to a uniform distribution within the appropriate lower/upper bounds in each dimension. This resulted in a total of 400 points for the four clusters. The clustering problem for both ART and LimitsART was to discover through adaptation to the 2-component input patterns the network weights that provide four templates, hence, four hyperboxes, one hyperbox to bound each of the four clusters.

LimitsART (where the simulator is given the threshold value $t > 0$, bringing the colimits and limits into play in the network architecture) with stack intervals restricts each side of its 2-dimensional hyperboxes to within the same allowed maximum length. Therefore, it would be expected to separate the four clusters exactly with the appropriate threshold value, $t = 0.8$, as long as the vigilance value ρ does not exceed 0.8. Since $1 - t = 0.2$, this threshold value allows a hyperbox to have a half-circumference(sum of lengths of the left and lower sides) of up to 0.4 units while restricting each side to a maximum length of 0.2. Therefore, a hyperbox cannot encompass points in more than one of the four well-separated clusters, yet is allowed to encompass all the points in a cluster. Generic ART (t = 0) with stack intervals has no such guarantee. With a vigilance value of $\rho = 0.8$, it generates hyperboxes which distribute the value 0.2 over all sides cumulatively, that is, a hyperbox might have one side at a length of 0.1 and the other at a length of 0.3, for a total half circumference of 0.4. Thus, a hyperbox might encompass points in two of the specified clusters while covering each cluster only partially.

With lower threshold settings, $0 < t < 0.8$, LimitsART would be expected to yield partial success, this effect might be enhanced by specifying values of ρ such that $0 < \rho < 0.8$, allowing the ART vigilance subsystem to provide a check on overall

hyperbox size. Parameter combinations such as these might well be used on arbitrary data sets, where the appropriate settings (if any) are not known and clustering with experimental parameter variation might be useful in a search for a combination yielding acceptable performance. Generic ART ($t = 0$) with $0 < \rho < 0.8$ would be expected to perform less well. If either parameter t or ρ exceeds 0.8, the points in each of the four clusters would be divided among several hyperboxes. This would obviate the identification of the correct clusters with single hyperboxes. Generic ART might still generate hyperboxes encompassing points in more than one cluster unless $\rho > 0.9$. In summary, it was expected that, with known, well-separated clusters, the generic architecture could (and often would) produce non-square hyperboxes that do not separate the four clusters appropriately ($\rho < 0.9, t = 0$) or else produce small hyperboxes that "fracture" the four clusters into smaller clusters ($\rho \geq 0.9$, $t = 0$). Since the sole objective was to test the LimitsART architectural modification, other clustering methods were not considered.

Several combinations of parameter values ρ, t were tried, with 20 runs made for each combination. The following procedure was used to obtain the input data for each combination over the 20 runs. First, the aforementioned procedure was used to obtain a new data set having four clusters each with 100 randomly located points lying within a hyperbox boundary as previously described. This data set was used for all 20 runs for that parameter combination. For each run, all 400 points from the data set were input to the stack interval/ART network simulator for clustering. The order in which the points were input was randomized anew for each run, and the points were input in the same order for each of the three passes through the data. Only two passes are necessary to stabilize the clusters for a data set whose input patterns are all of a single size-see the seminal paper on ART clustering by Heileman et al. [65]. The third pass was used for an automated procedure that tallies the results due to the classifications not being stable until the end of the second pass. The stack interval coding for converting the two real values in each input pattern

into a binary pattern was as follows. The positive stack for each of the $n = 2$ real values consisted of $N_{psn} = 100$ binary bits (the notation used here is the same as in previous sections). With the complementary stacks included, this yielded a size of $N = n \cdot 2N_{psn} = 2 \cdot 2N_{psn} = 4N_{psn} = 400$ bits in each of the resulting input patterns for ART, which therefore had 400 input $F0$ nodes and 400 $F1$ nodes. For each combination of network parameters ρ, t, the data set with its 400 2-dimensional points was converted in this fashion by the stack interval networks "on the fly" to 400 ART binary input patterns during each pass through the data. The parameter value combinations used were selected to simulate experimentation with an arbitrary data set to find a combination yielding acceptable (loosely, "the best") performance.The fact that the correct hyperboxes were known in advance for this experiment allowed us to develop a figure of merit for measuring performance with a single number. We call this number "Efficiency", defined as follows. First, we analyzed the result of each run for each parameter combination to find the number "Correct Templates" for that run. A correct template is one whose hyperbox encompasses exactly the points in one of the four clusters, if a hyperbox either spans more than one cluster or is restricted to one cluster but omits some of its points, the template is incorrect. We also obtained the number "Total Templates" for that run, and then found "Efficiency" as the quotient $Efficiency = \frac{CorrectTemplates}{TotalTemplates}$. Obviously, perfect performance is signified by an efficiency value of 1.0. Finally, we defined the number "Abandoned Templates" to keep track of templates which, in the final pass through the data for one run, "lose" all the points which were assigned to the template during earlier passes.

E.2 Experimental Results

Figure E.3 and Figure E.4 show the results for one run each at the indicated parameter combinations. In the result shown, stack interval/ART($t = 0$) with $\rho = 0.8$

Parameters		Means (standard deviations)			
ρ	t	Total templates	Abandoned templates	Correct templates	Efficiency
0.7	0.0	7.4 (1.1)	0.05 (0.22)	0.3 (0.47)	0.04 (0.07)
0.7	0.65	5.25 (0.97)	0.0 (0.0)	2.0 (1.3)	0.44 (0.36)
0.75	0.0	7.75 (2.38)	0.55 (0.76)	1.9 (1.2)	0.31 (0.26)
0.75	0.65	5.35 (1.09)	0.1 (0.3)	2.7 (1.1)	0.56 (0.31)
0.75	0.7	4.4 (0.50)	0.05 (0.22)	3.3 (1.0)	0.78 (0.29)
0.8	0.0	4.75 (0.72)	0.4 (0.5)	3.45 (0.89)	0.76 (0.26)
0.8	0.7	4.3 (0.47)	0.15 (0.37)	3.7 (0.73)	0.88 (0.22)
0.8	0.75	4.1 (0.31)	0.0 (0.0)	3.8 (0.62)	0.94 (0.18)
0.8	0.8	4.0 (0.0)	0.0 (0.0)	4.0 (0.0)	1.0 (0.0)
0.9	0.0	23.5 (1.4)	0.05 (0.22)	0.0 (0.0)	0.0 (0.0)
0.9	0.8	23.5 (1.4)	0.05 (0.22)	0.0 (0.0)	0.0 (0.0)

Figure E.5: Summary of LimitsART experiment results. Note that the cases with $t = 0.0$ are results for generic ART 1. Nonzero values of t apply to LimitsART.

(the optimal value for t) yields one abandoned template and one that spans two clusters, the two templates that exactly encompass the clusters on the left are correct. Thus, there are two correct templates and six total templates, with one abandoned. Figure E.5 shows the results for all parameter combinations used in this experiment, performed as if increasing values were being tried, with ρ always greater than t(otherwise, vigilance is completely overridden by the effect of the threshold value). The aggregate results for template counts and efficiencies are shown as means and standard deviations over the 20 runs for each combination. Notice that for each value of ρ, the results vary systematically in favor of increasing performance with increasing t. All result reflecting numbers show this trend; Total Templates, Abandoned Templates, Correct Templates, and Efficiency. As either ρ or t approaches

the value 0.8 from below, performance increases,with increasing t having the more dramatic effect for $\rho \leq 0.75$. For $\rho = 0.8$, the efficiency increases with t toward unity, reaching it only at $t = 0.8$, where the effect of ρ is completely overridden. At the two parameter combinations tried for $\rho = 0.9$, the efficiency drops to zero regardless of whether $t = 0$. Efficiency as we have defined it imposes a rather stringent performance measure, but the number Total Templates does suggest a sudden increase in the number of templates with a falling-off in performance for ρ values exceeding 0.8. This effect might be noticed with an arbitrary dataset using some other criterion for performance, based upon the observed effects of clustering in the domain from which the data were obtained. In summary, the results show clearly the intended effect in an experiment designed for this purpose of evaluating LimitsART. Given that LimitsART performs hyperbox control as intended, the category theory described earlier implies that LimitsART is capable of representing categorical limit structures. As such, LimitsART will be incorporated into the embodied computational neural architecture.

Appendix F

Behavior Measures

F.1 Single Cardinal Releases

This section contains results from experiments where the rat has no prior experience. That is, there are no learned templates in the model brain. The rat is released, and then returned to the release point each time the platform is located. Each set of releases, where the brain starts with physical structure but no conceptual structure, then forms sensory-motor memory bindings until the platform is repeatably located, is called an experiment. Within each experiment, the behavior from one release until the platform is found is called a trial.

For each experiment, four graphs are displayed. The first graph shows pathlength per trial. That is, how far the rat traveled from the release point until the platform was located. It is evident in the pathlength graphs when the rat has learned a path. After a path is learned in full, subsequent recalled motor sequences will follow this same path, so the pathlength will repeat.

F.1.1 North release

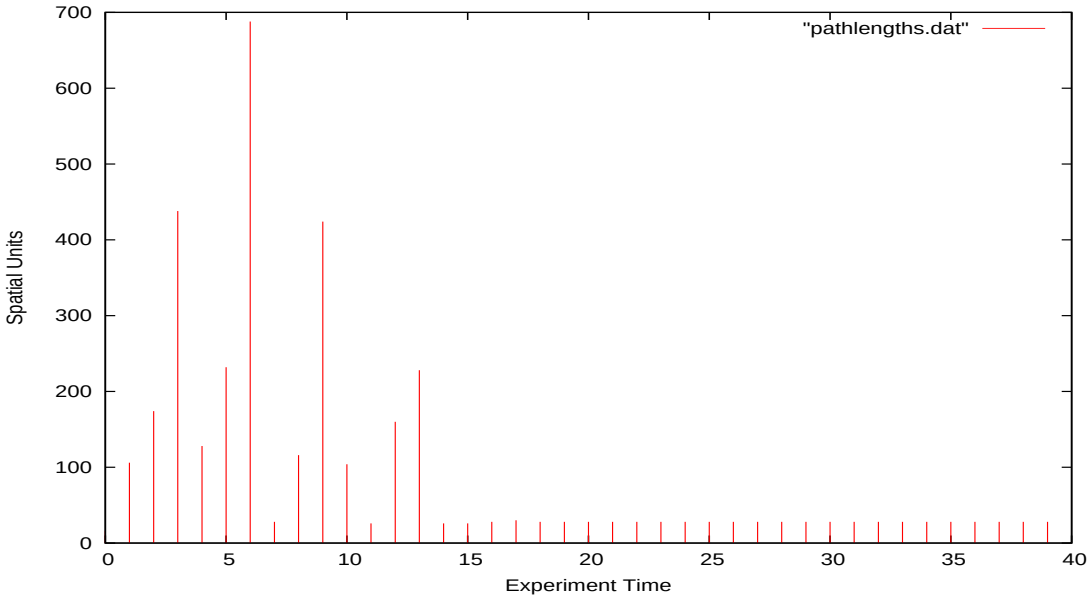

Figure F.1: Pathlengths from North release point.

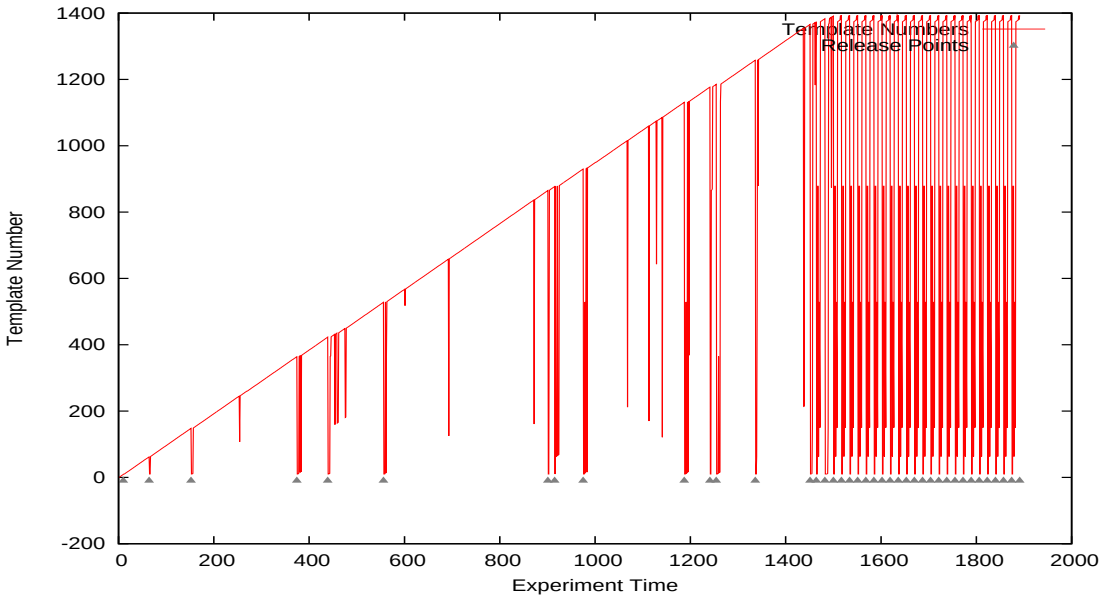

Figure F.2: Distal template formation from North release point.

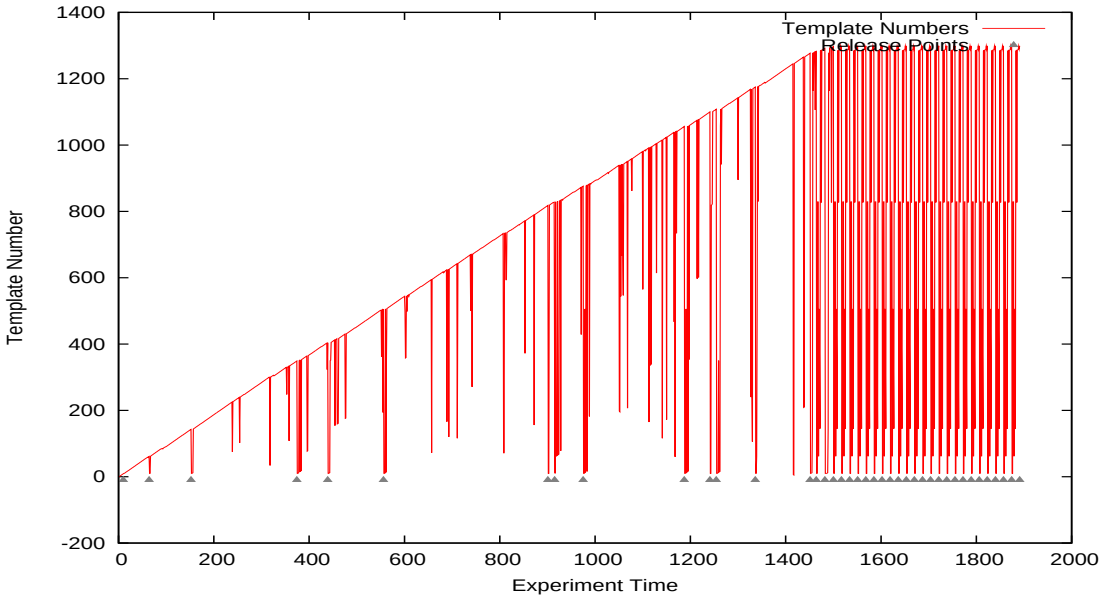

Figure F.3: Proximal template formation from North release point.

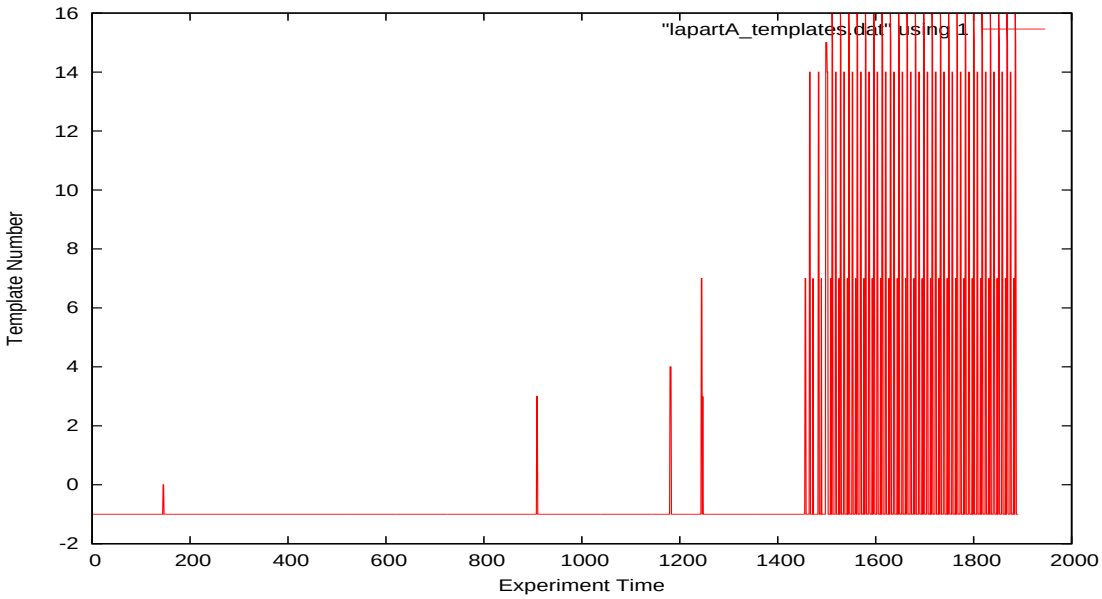

Figure F.4: LAPART A template formation from North release point.

F.1.2 South release

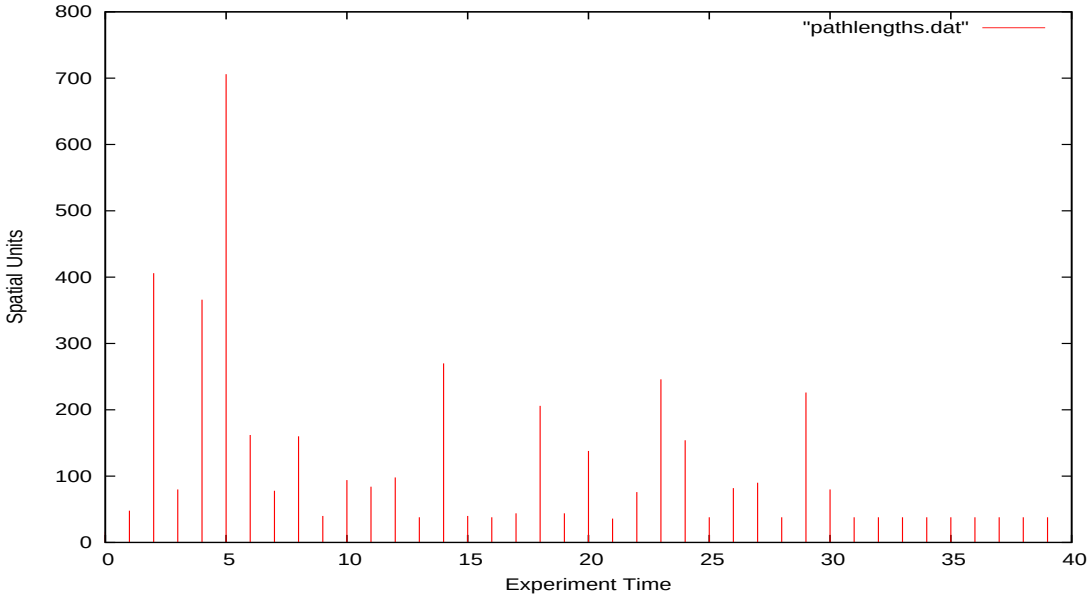

Figure F.5: Pathlengths from South release point.

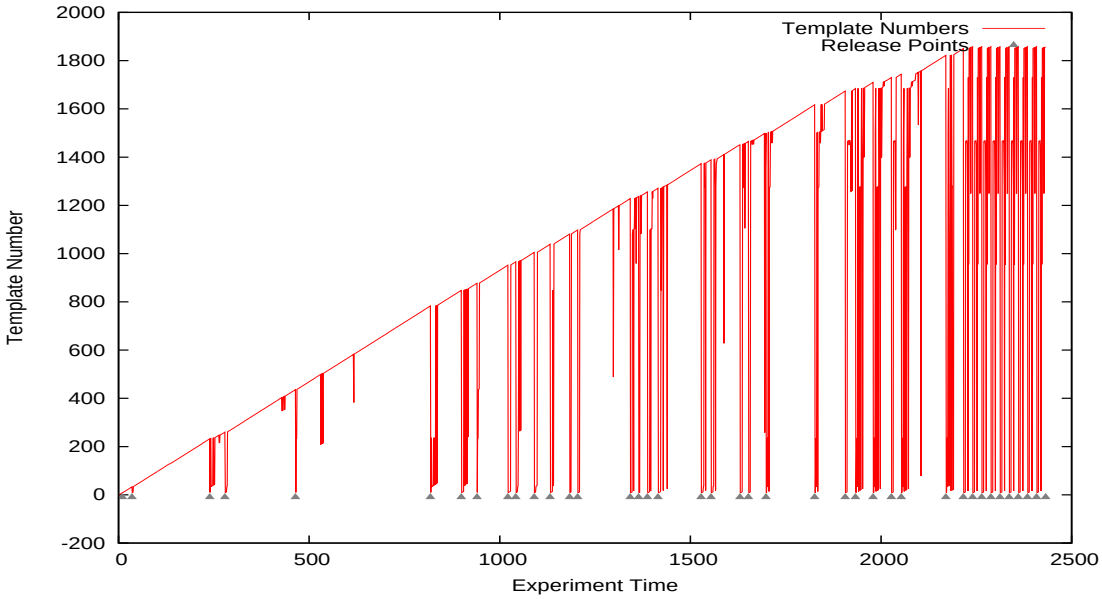

Figure F.6: Distal template formation from South release point.

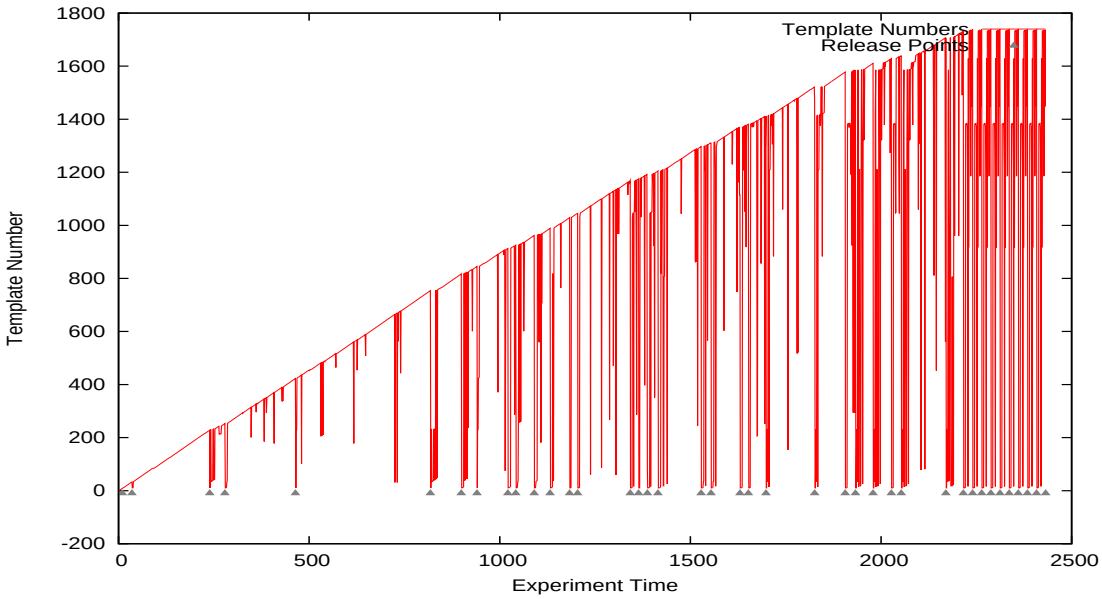

Figure F.7: Proximal template formation from South release point.

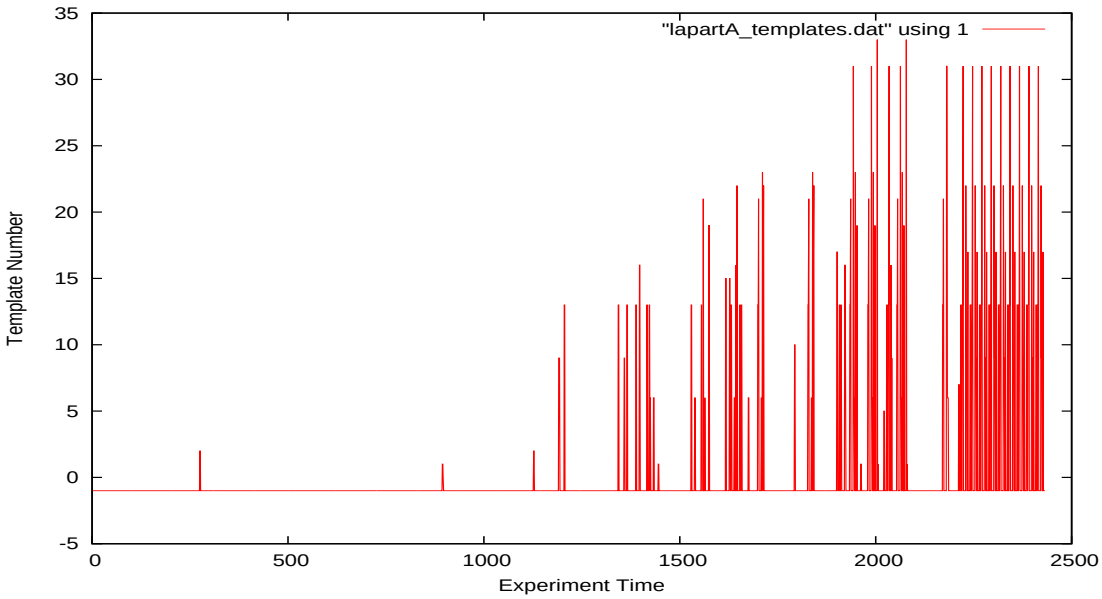

Figure F.8: LAPART A template formation from South release point.

F.1.3 East release

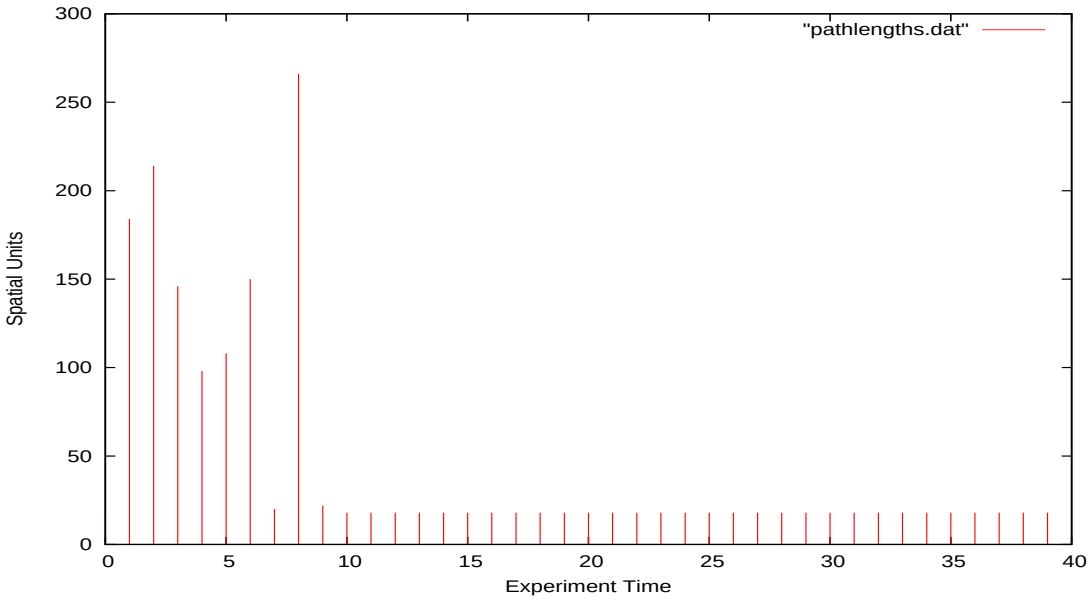

Figure F.9: Pathlengths from East release point.

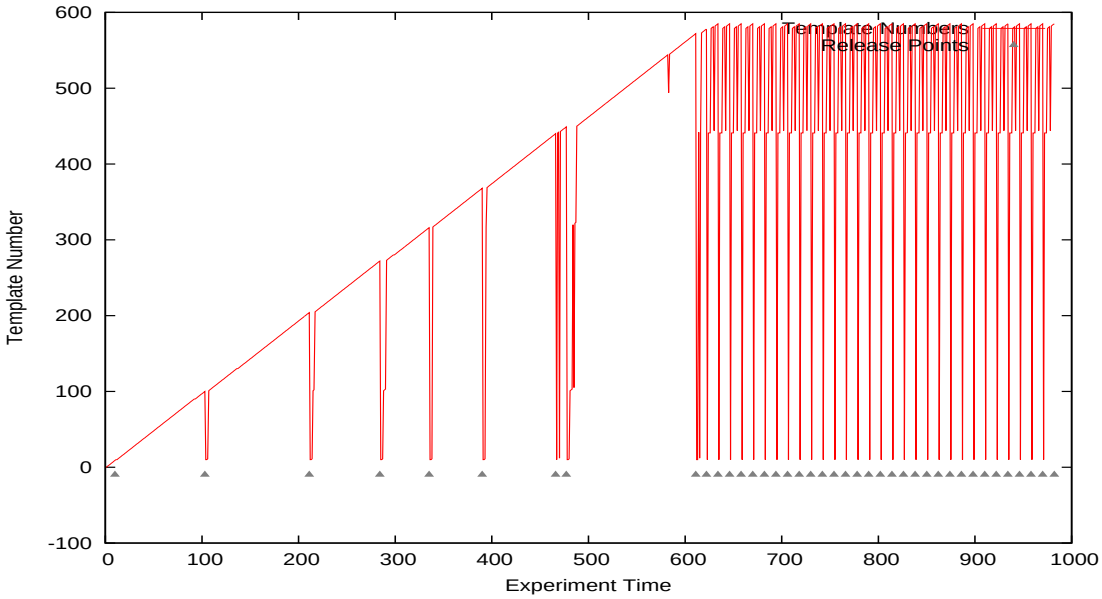

Figure F.10: Distal template formation from East release point.

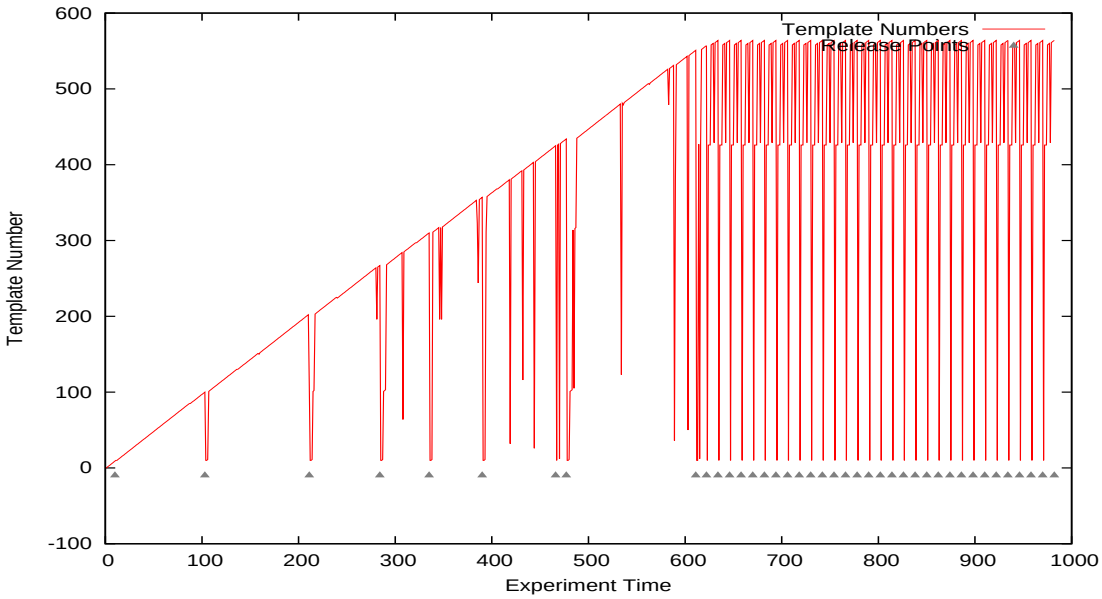

Figure F.11: Proximal template formation from East release point.

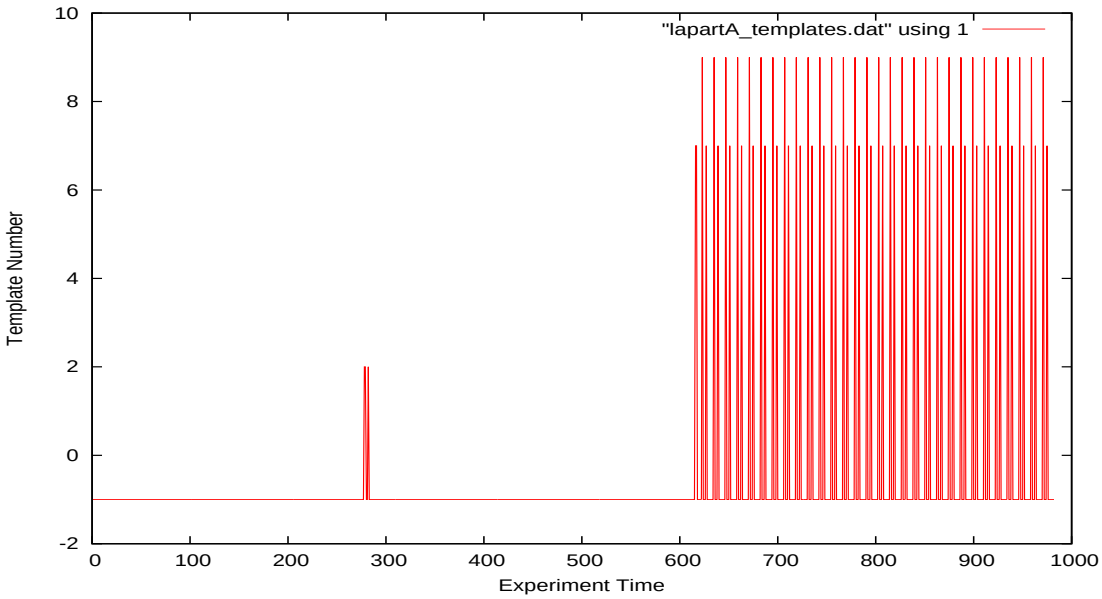

Figure F.12: LAPART A template formation from East release point.

F.1.4 West release

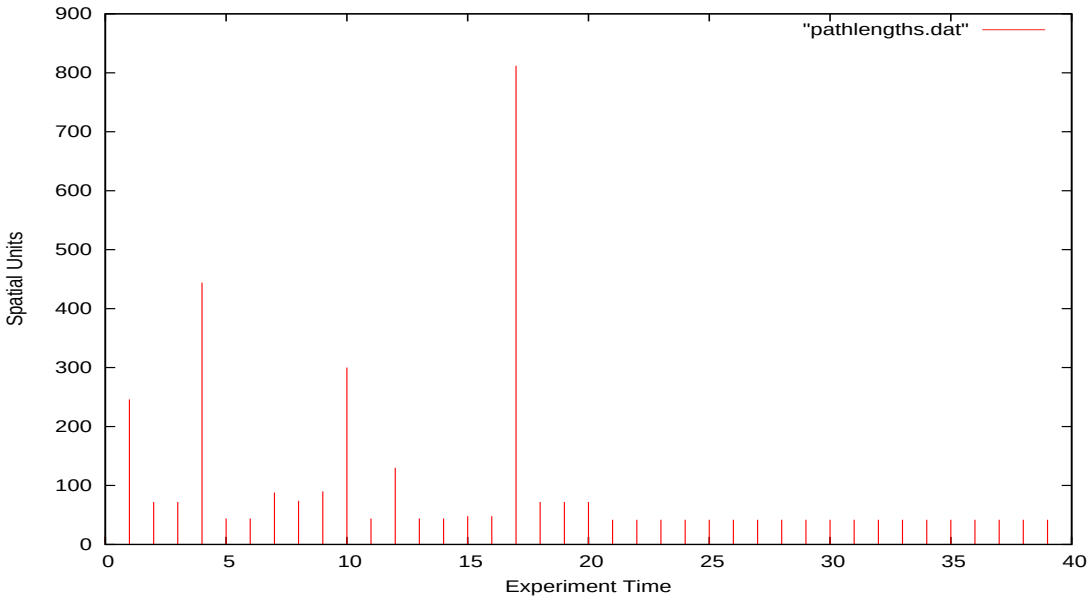

Figure F.13: Pathlengths from West release point.

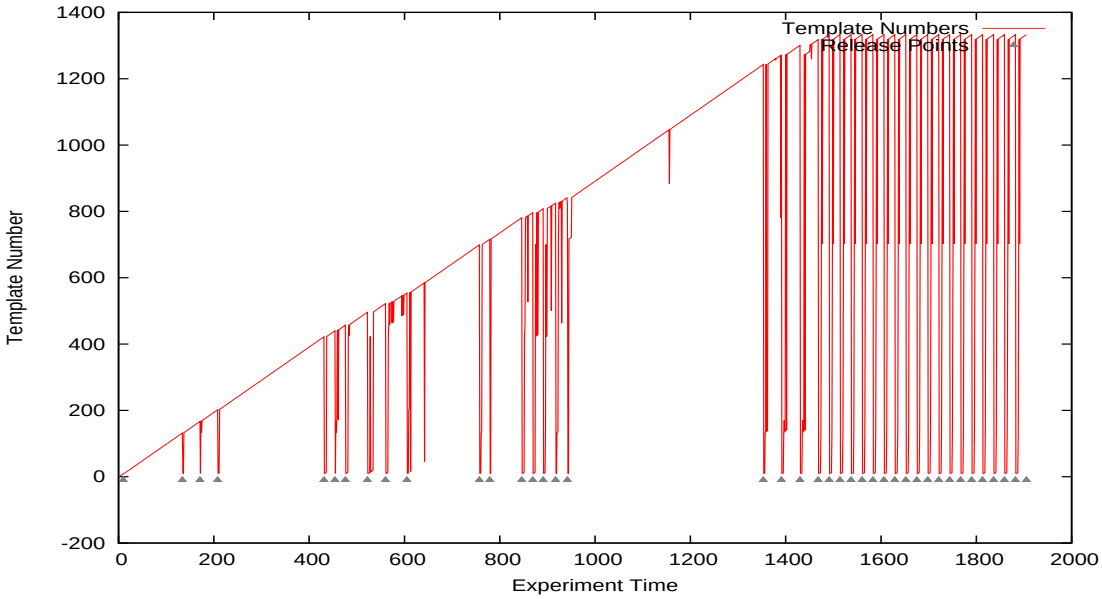

Figure F.14: Distal template formation from West release point.

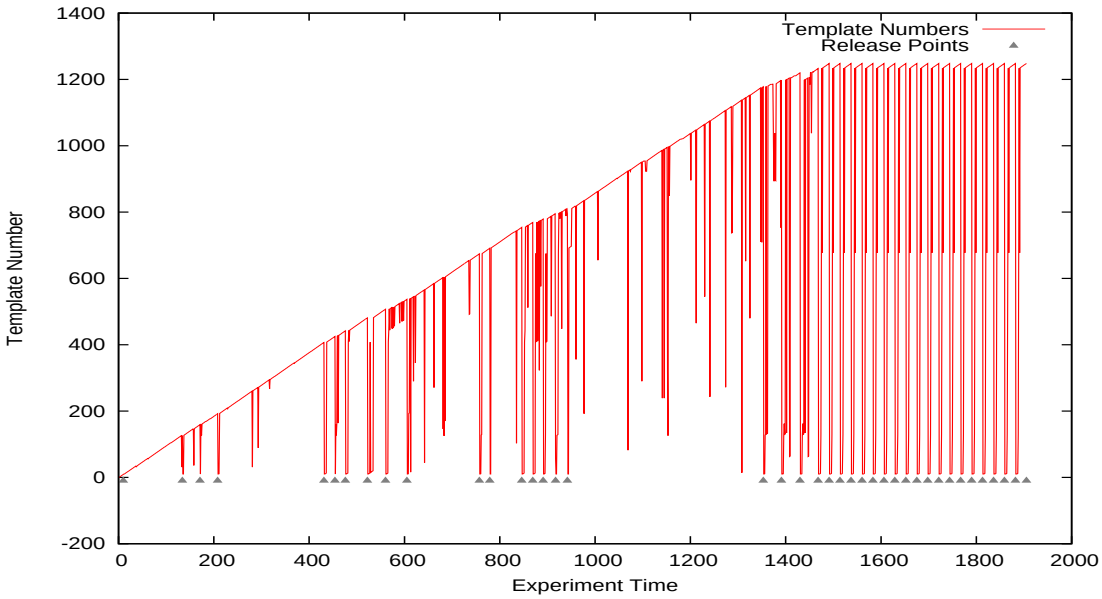

Figure F.15: Proximal template formation from West release point.

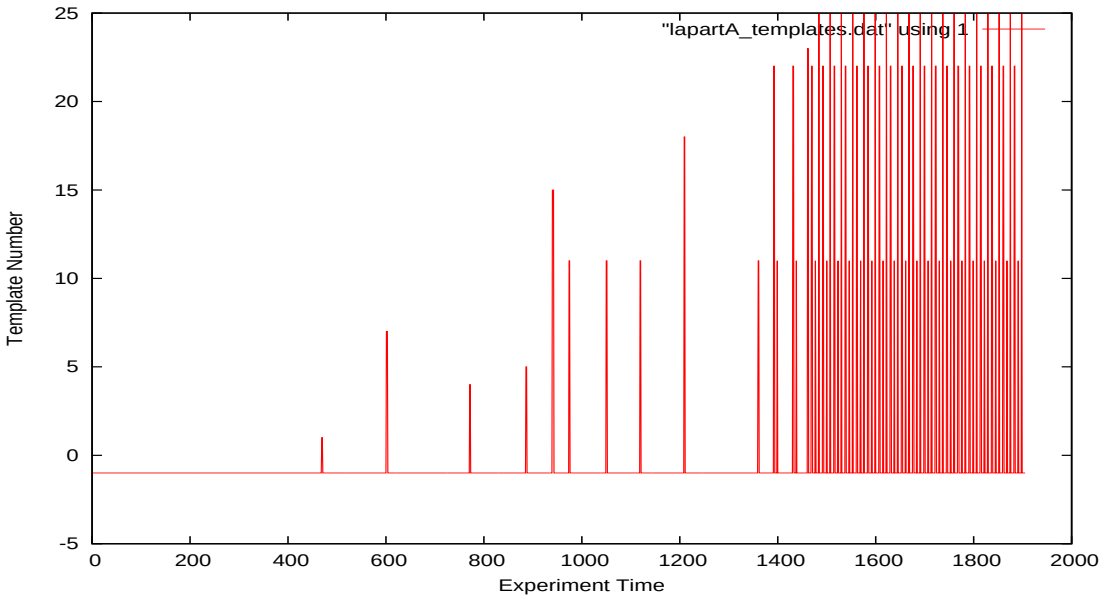

Figure F.16: LAPART A template formation from West release point.

F.2 Prior Experience Releases

This section contains results from experiments where the rat has previously learned to navigate to the platform from a single release point. Then for the following graphs, the rat is released from a different point. The rat is released, and then returned to the release point each time the platform is located. For each set of releases, when the computational brain starts with physical structure but no conceptual structure, then forms sensory-motor memory bindings until the platform is repeatably located, this is called an experiment. Within each experiment, the behavior from one release until the platform is found is called a trial.

F.2.1 North then West release

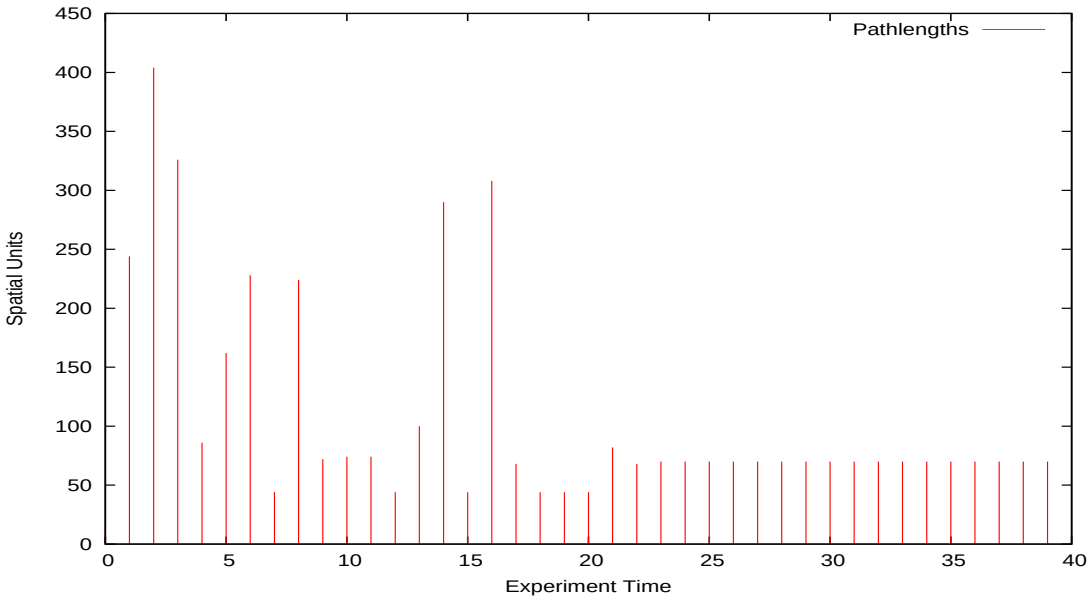

Figure F.17: Pathlengths from NW release point.

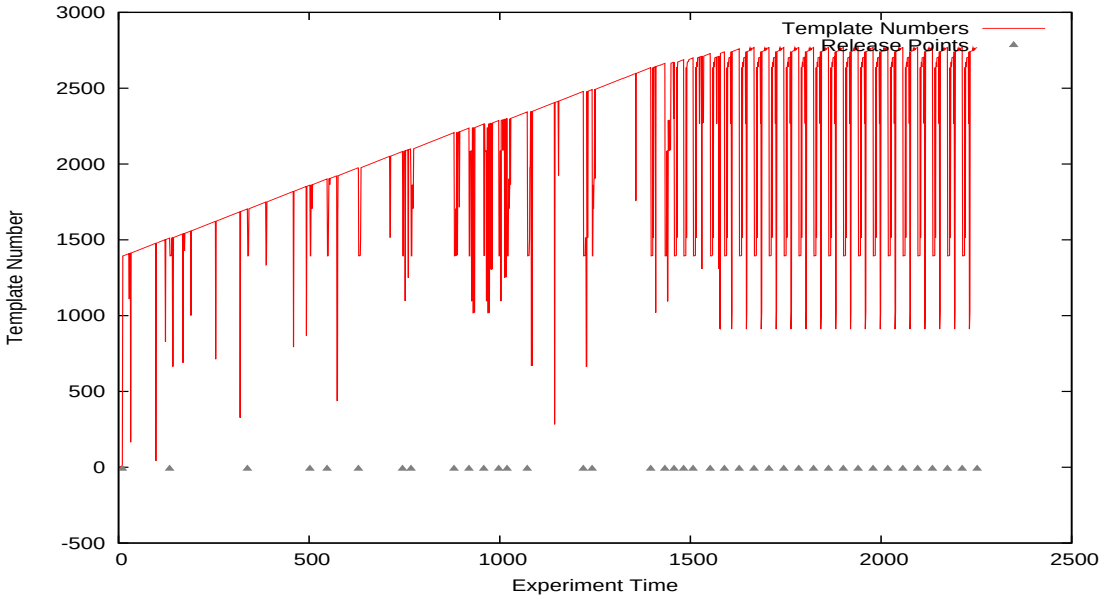

Figure F.18: Distal template formation from NW release point.

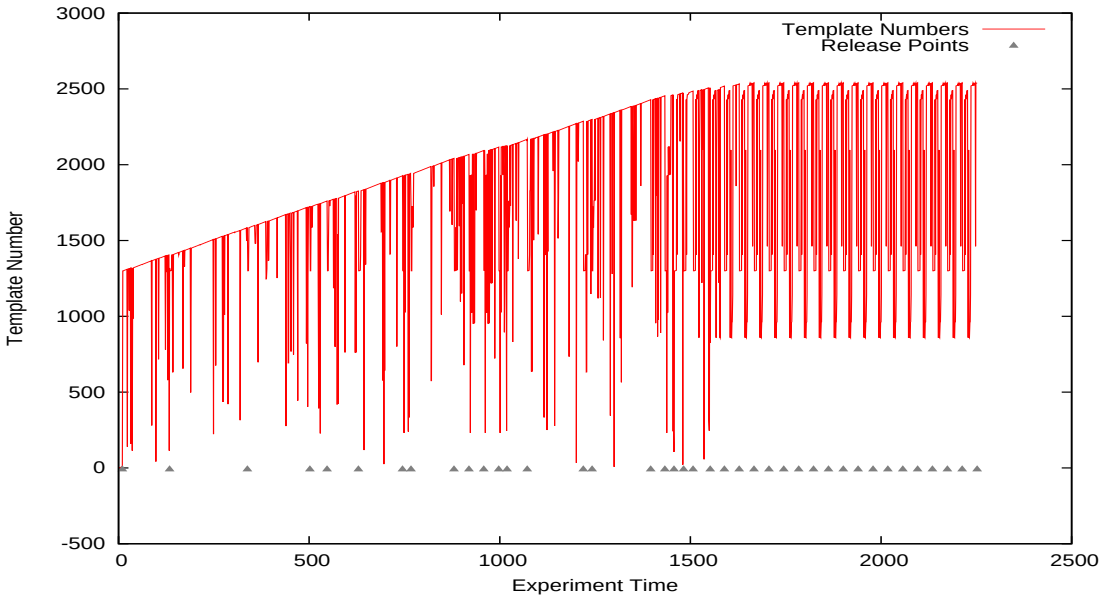

Figure F.19: Proximal template formation from NW release point.

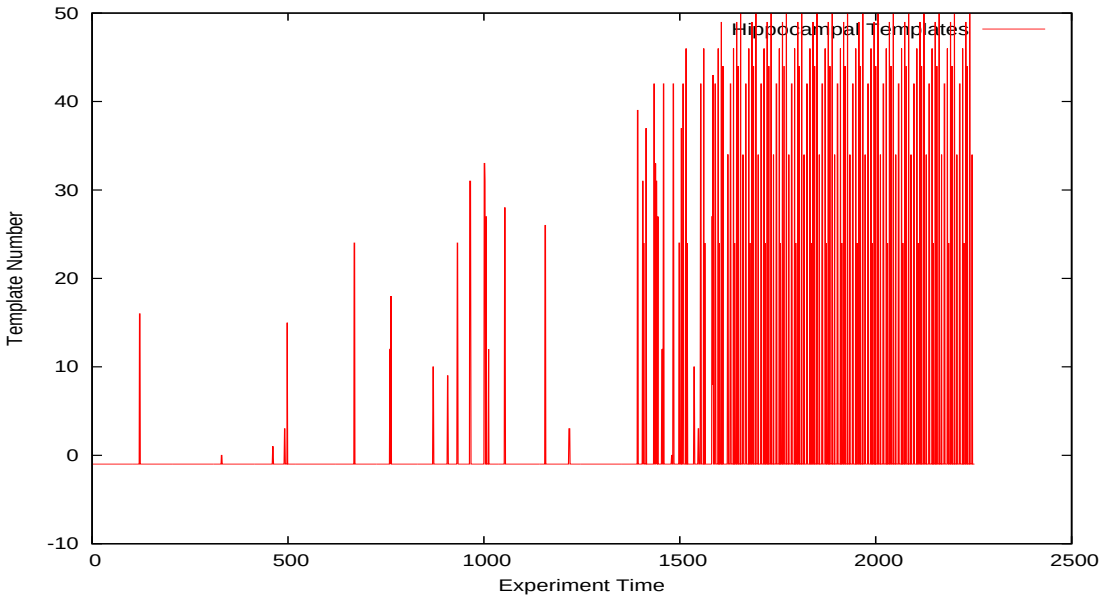

Figure F.20: LAPART A template formation from NW release point.

F.2.2 West then South release

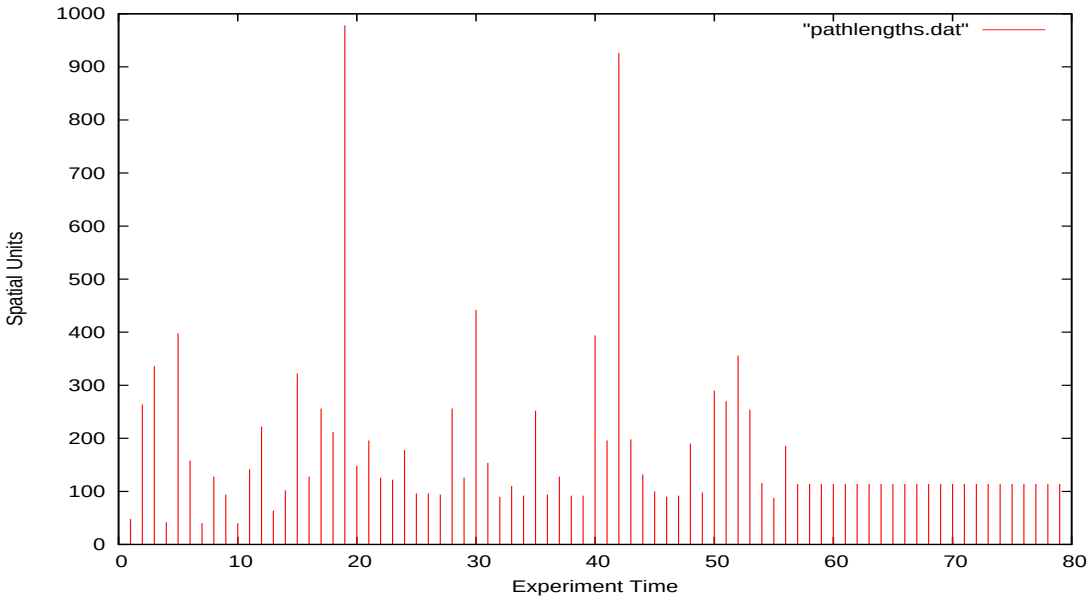

Figure F.21: Pathlengths from WS release point.

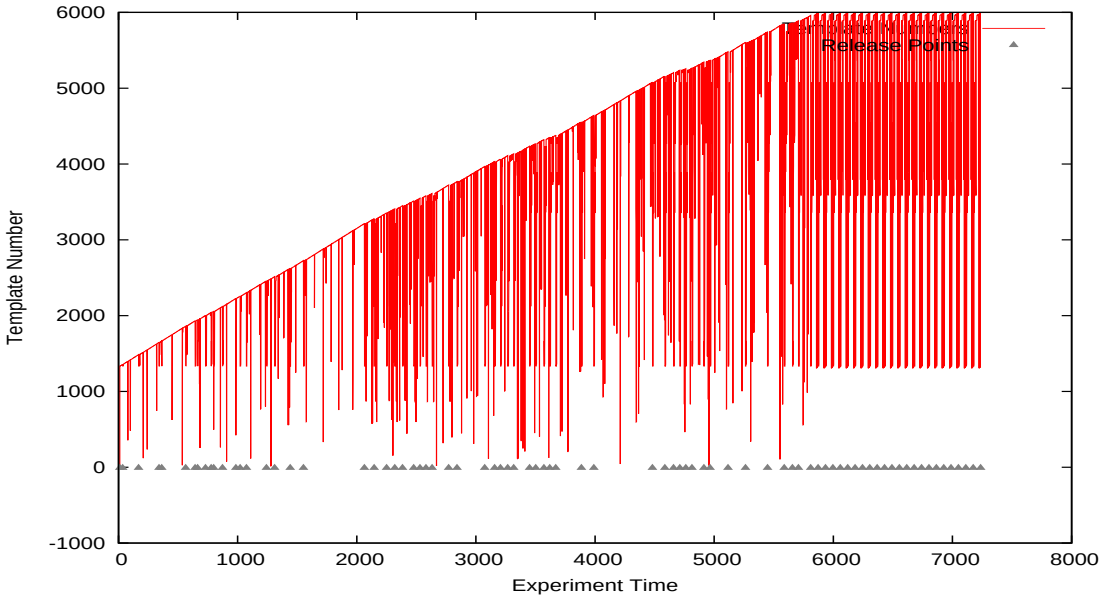

Figure F.22: Distal template formation from WS release point.

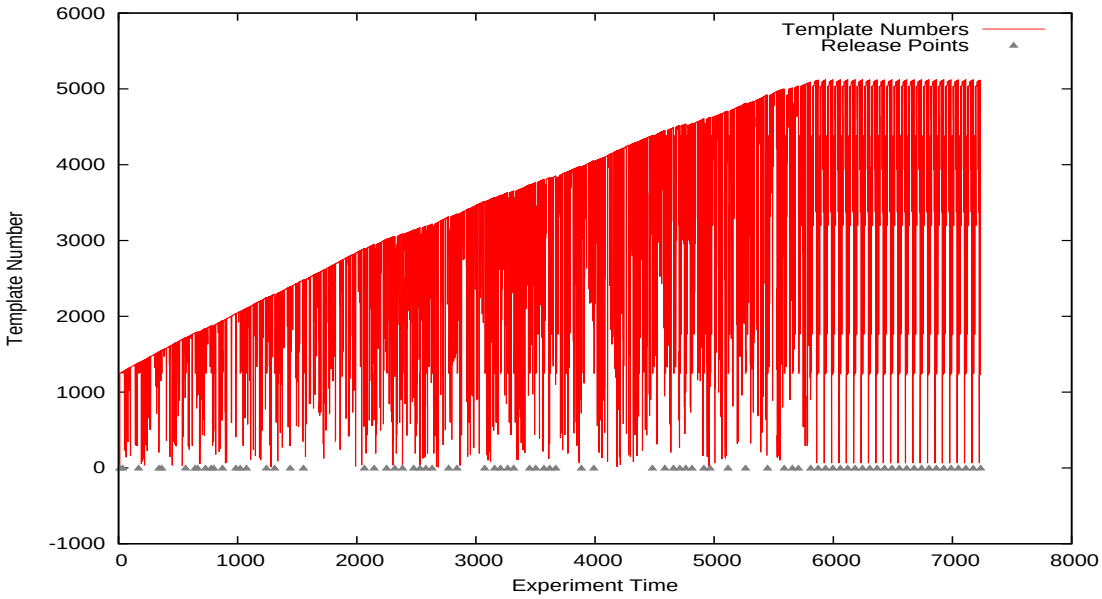

Figure F.23: Proximal template formation from WS release point.

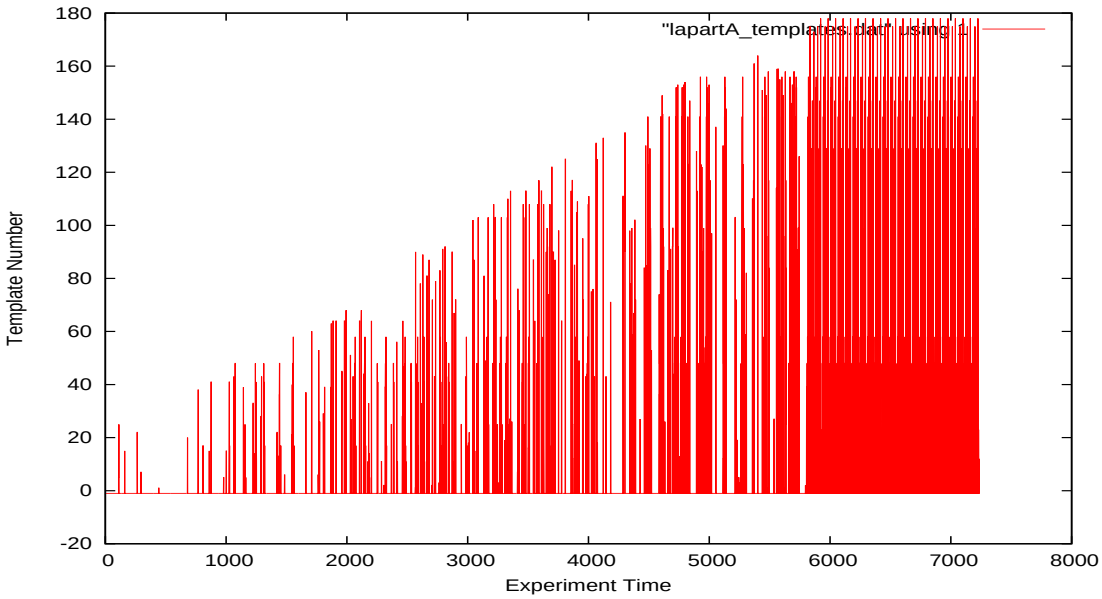

Figure F.24: LAPART A template formation from WS release point.

F.2.3 South then East release

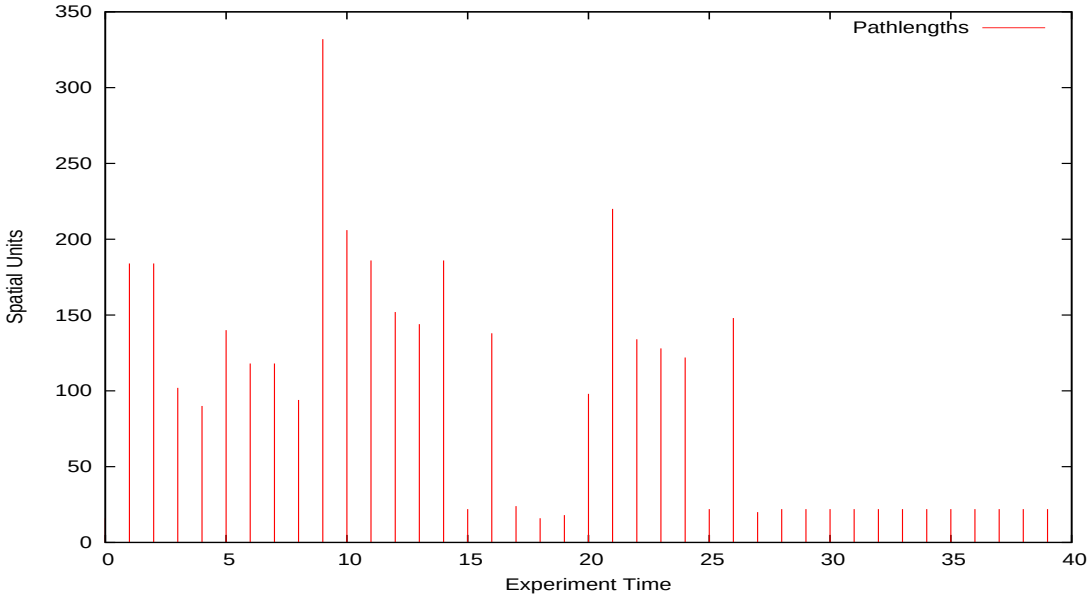

Figure F.25: Pathlengths from SE release point.

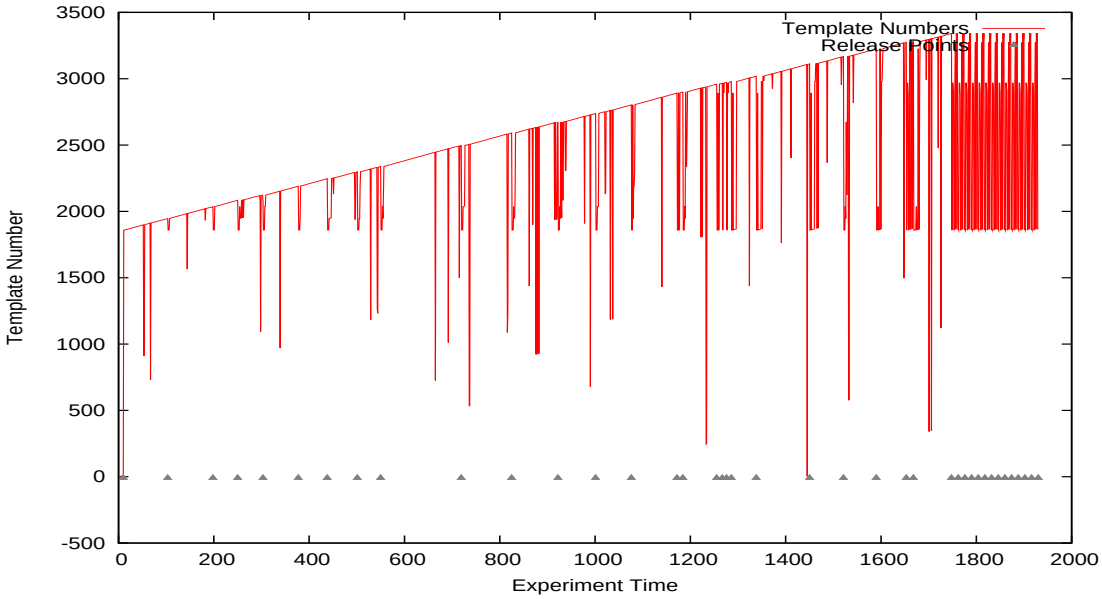

Figure F.26: Distal template formation from SE release point.

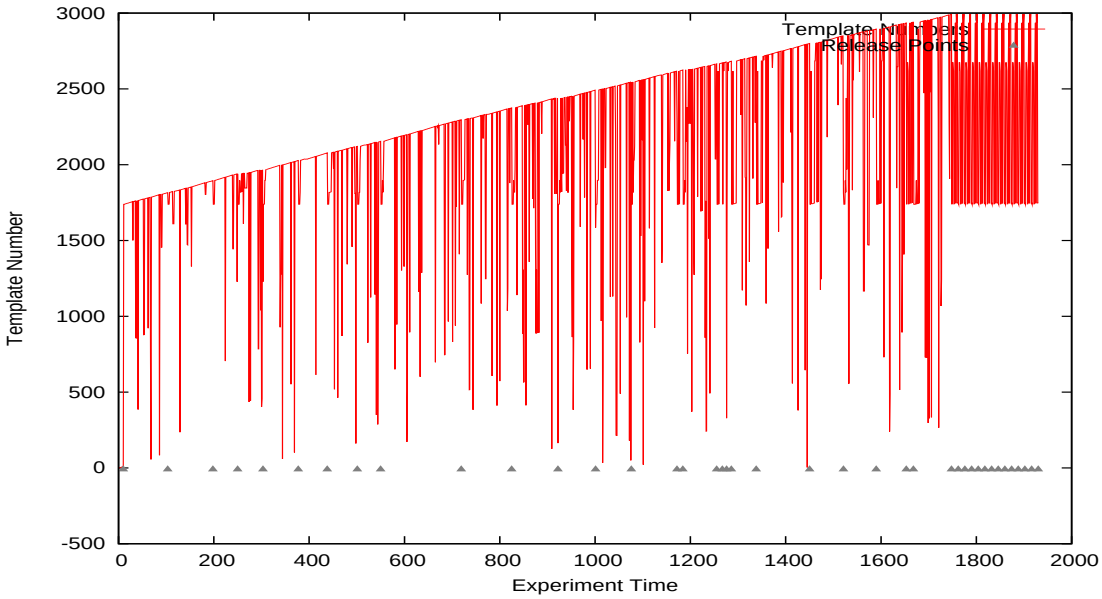

Figure F.27: Proximal template formation from SE release point.

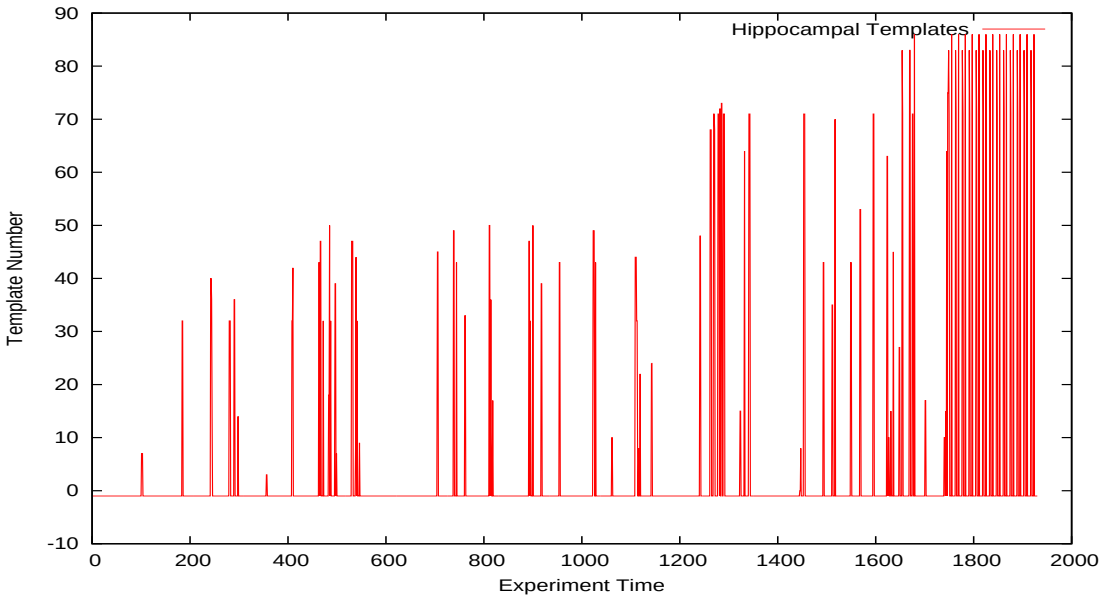

Figure F.28: LAPART A template formation from SE release point.

F.2.4 East then North release

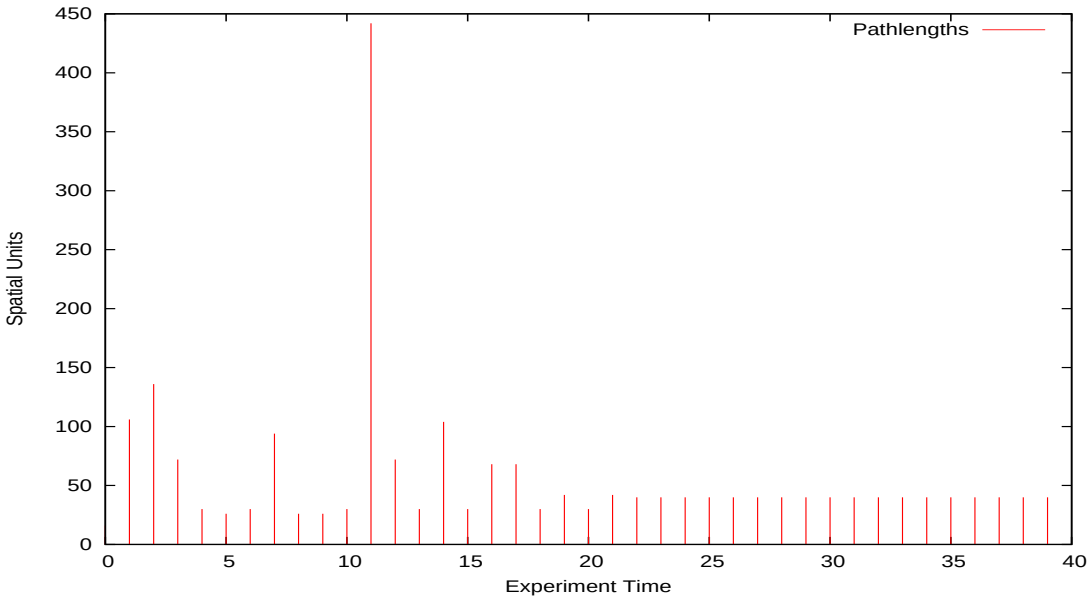

Figure F.29: Pathlengths from EN release point.

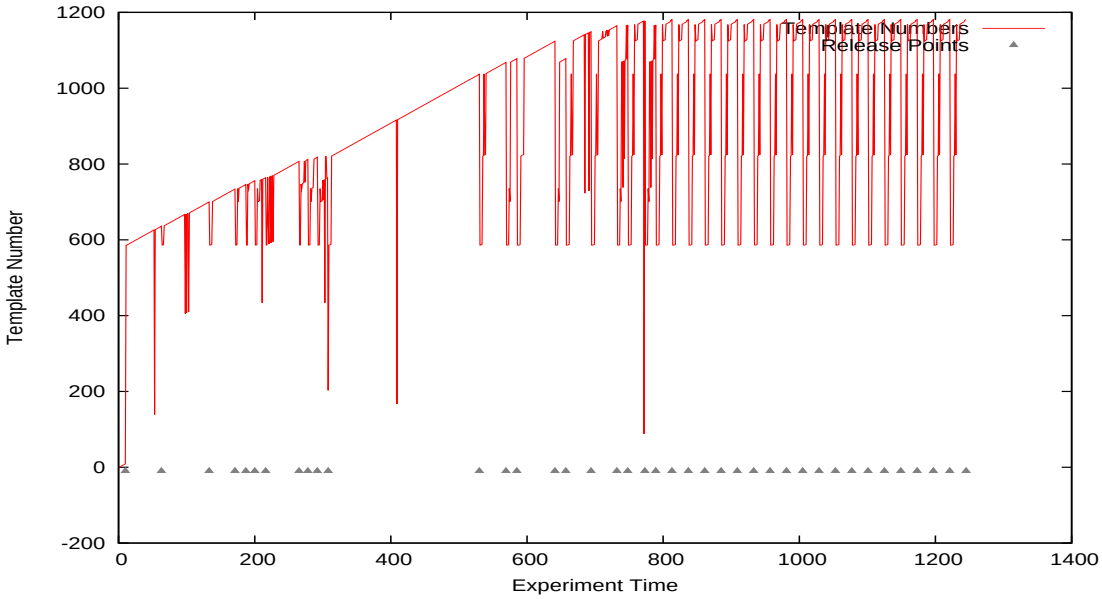

Figure F.30: Distal template formation from EN release point.

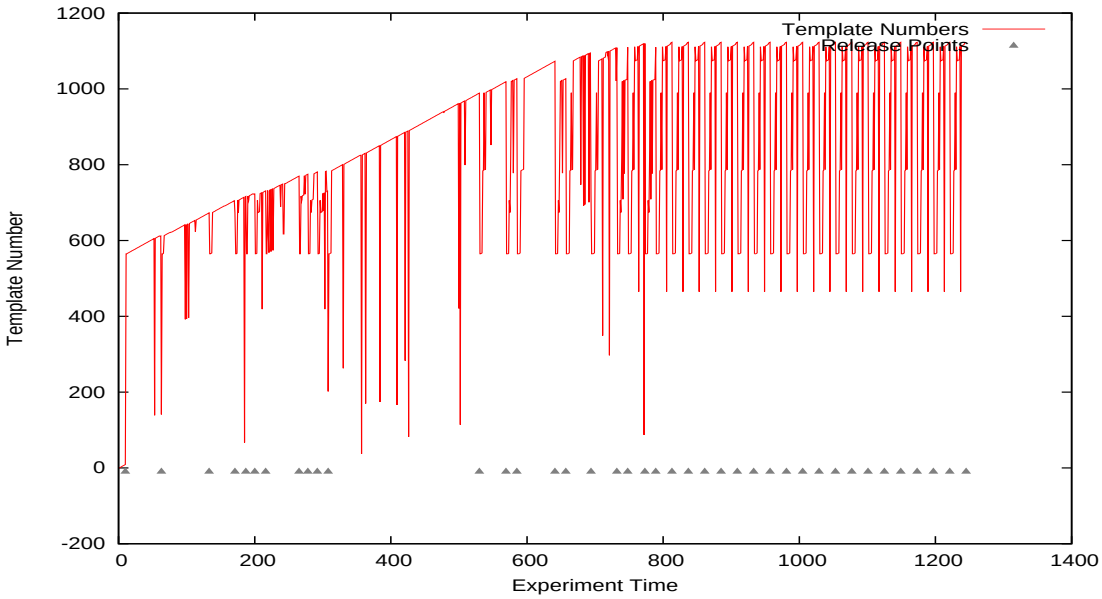

Figure F.31: Proximal template formation from EN release point.

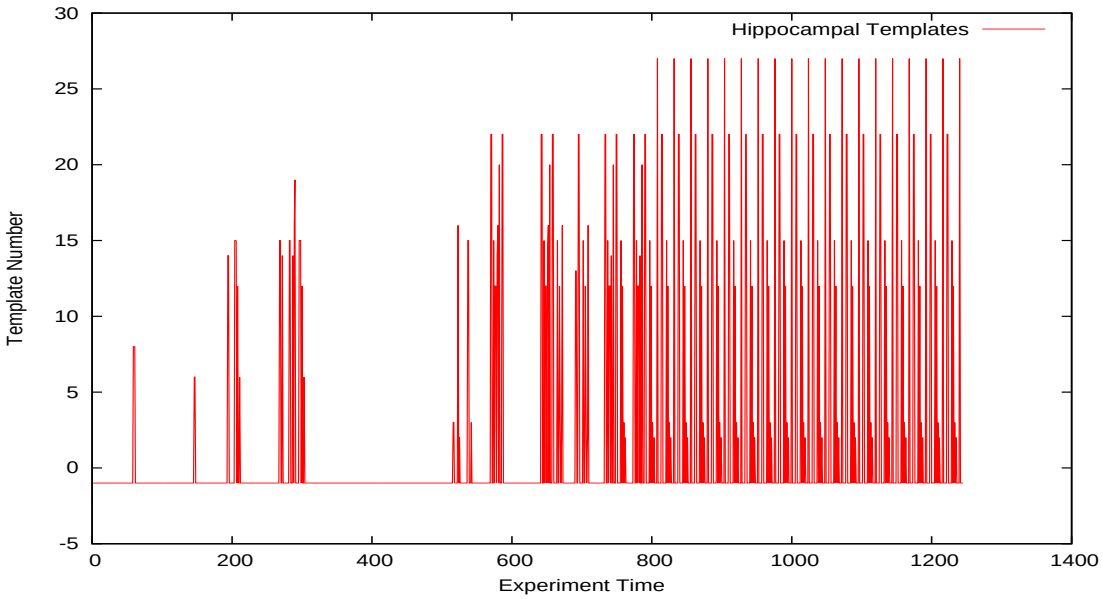

Figure F.32: LAPART A template formation from EN release point.

F.2.5 North then East release

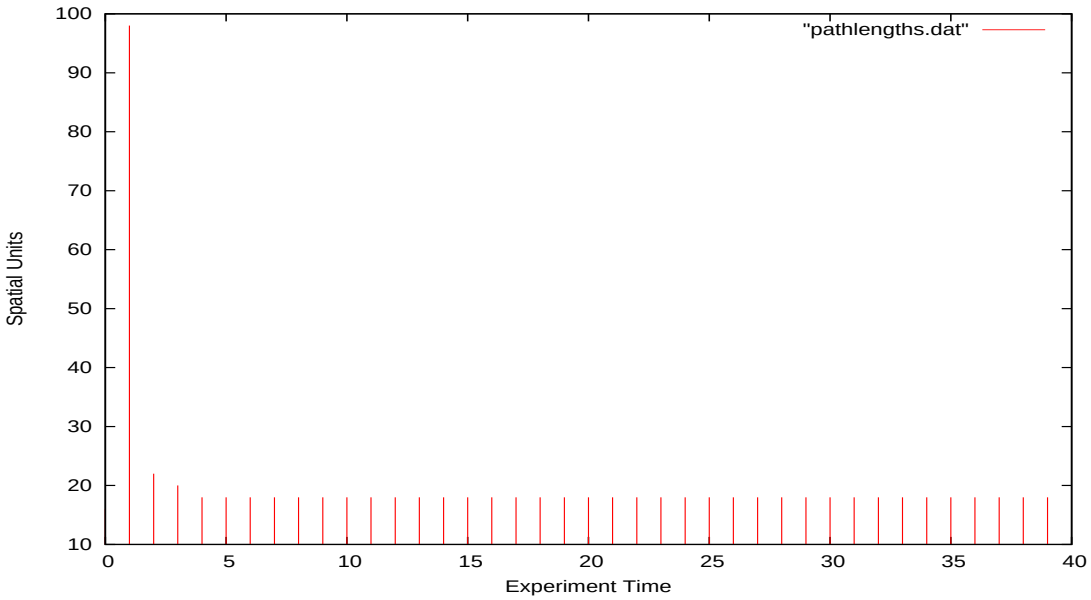

Figure F.33: Pathlengths from NE release point.

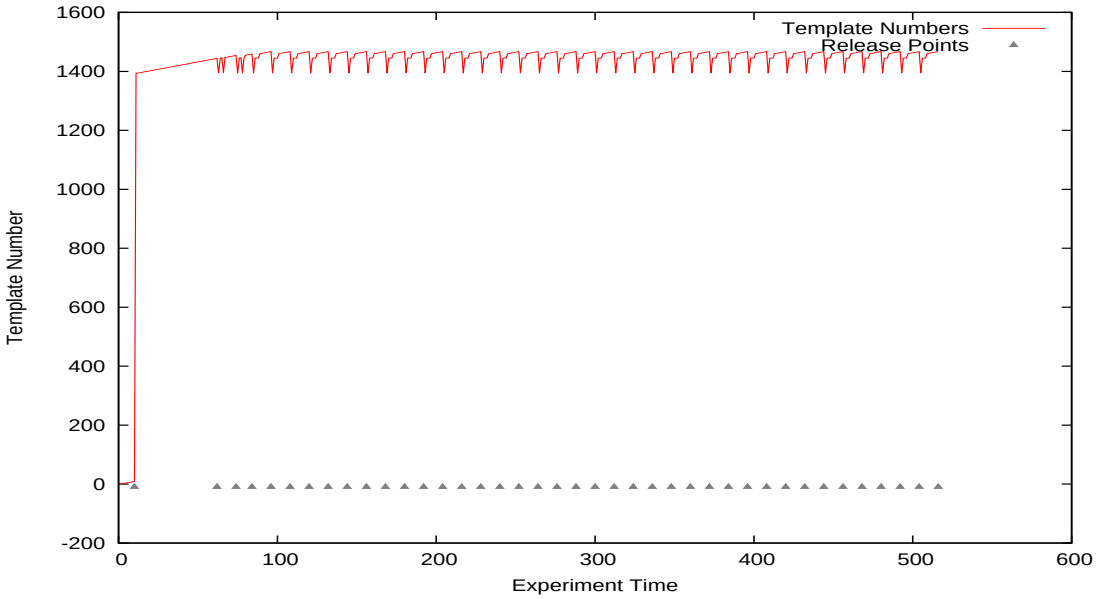

Figure F.34: Distal template formation from NE release point.

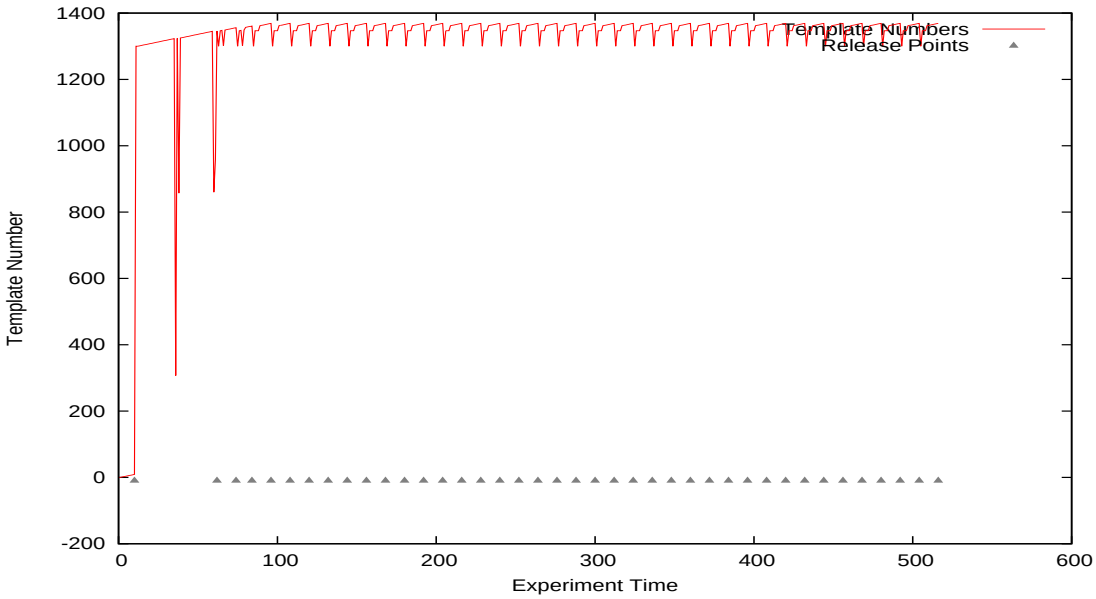

Figure F.35: Proximal template formation from NE release point.

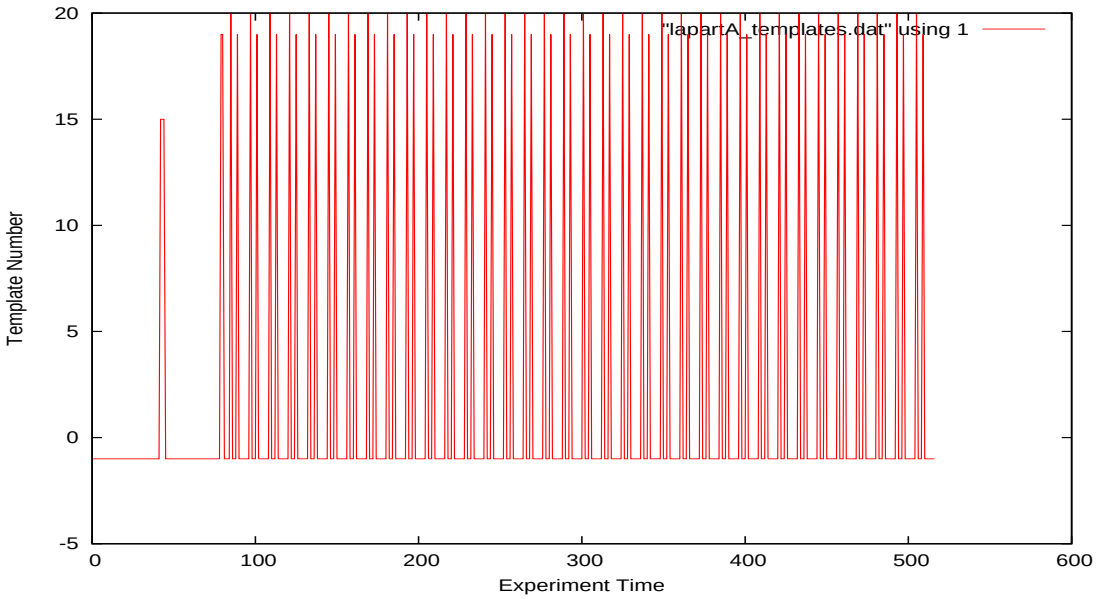

Figure F.36: LAPART A template formation from NE release point.

F.2.6 East then South release

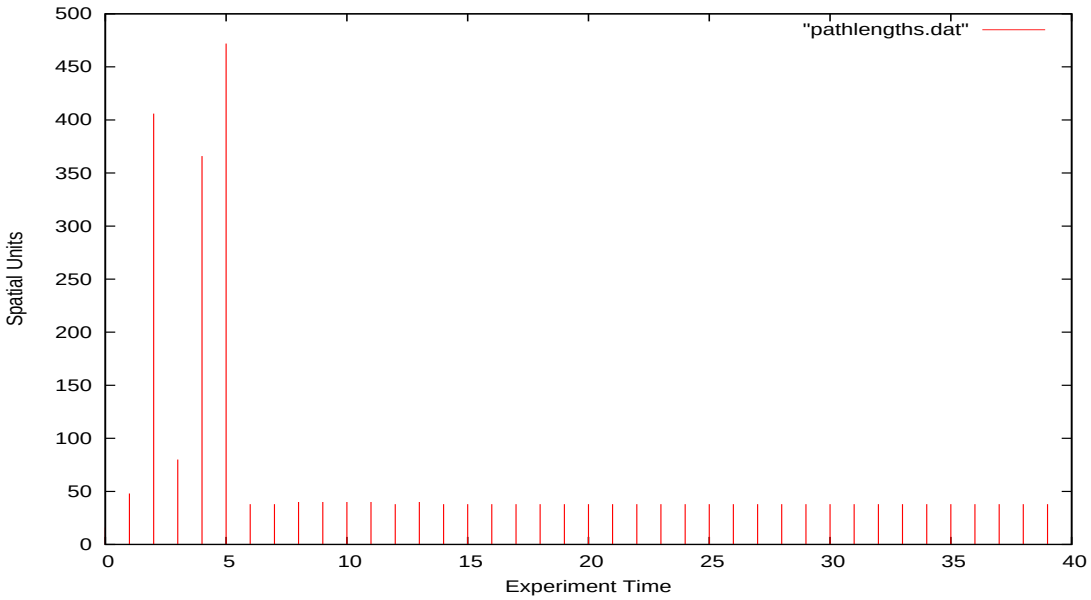

Figure F.37: Pathlengths from ES release point.

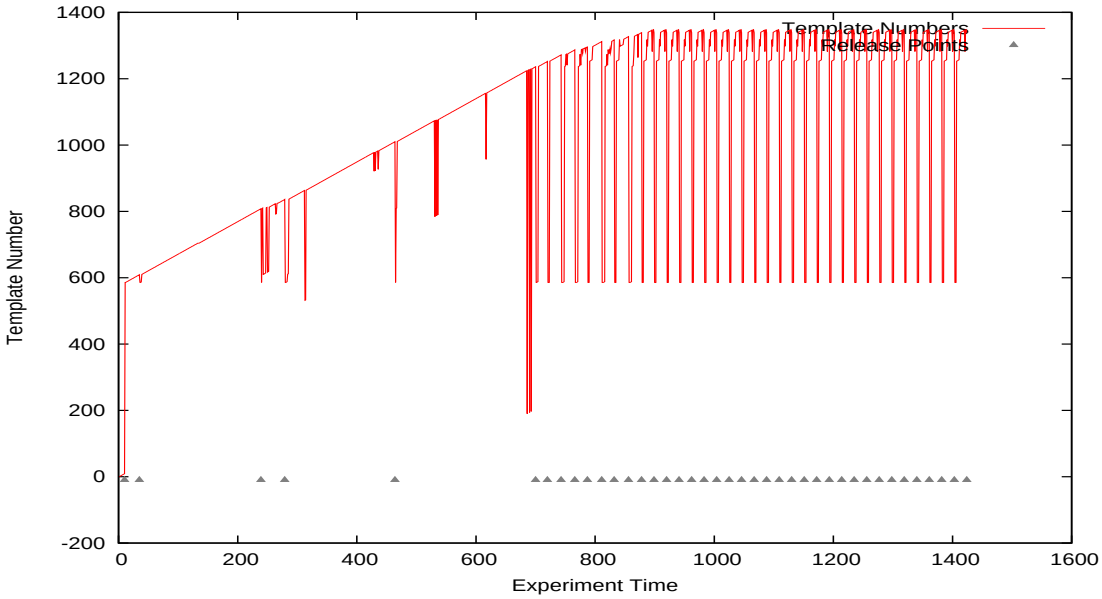

Figure F.38: Distal template formation from ES release point.

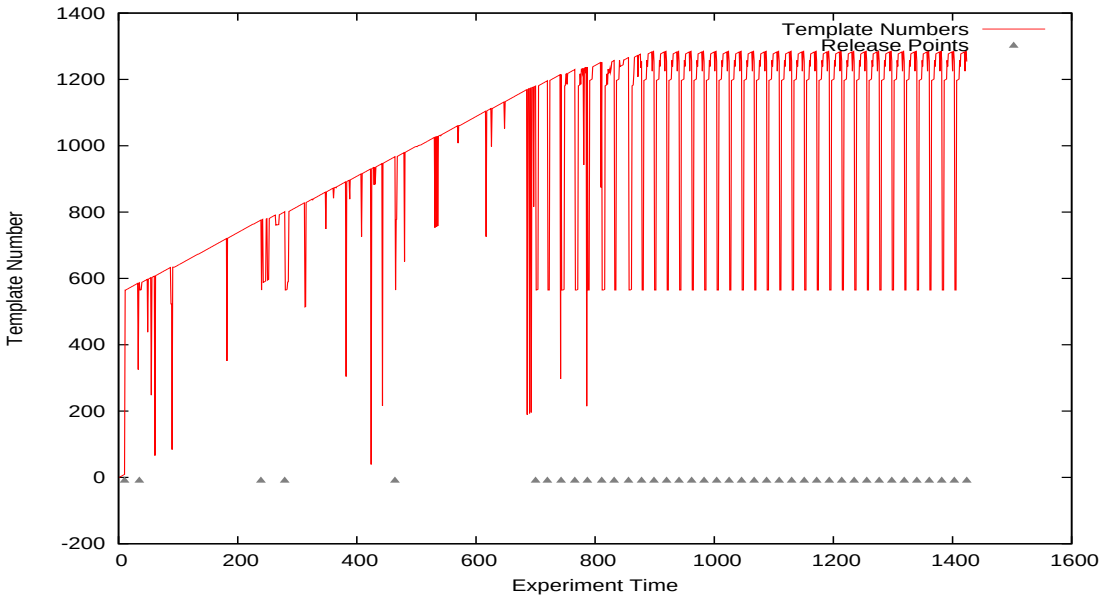

Figure F.39: Proximal template formation from ES release point.

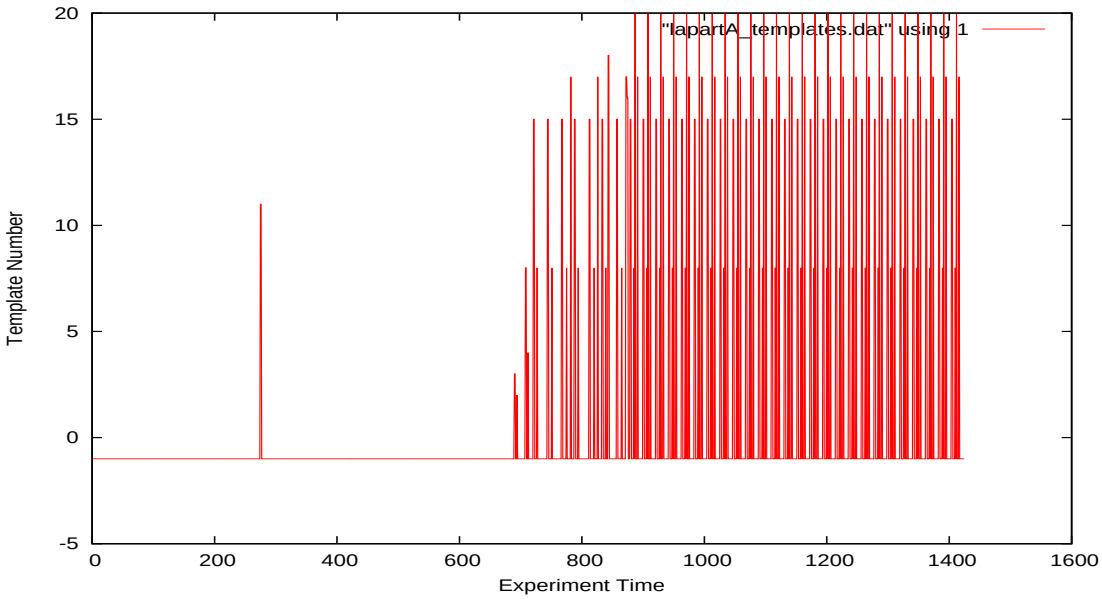

Figure F.40: LAPART A template formation from ES release point.

F.2.7 South then West release

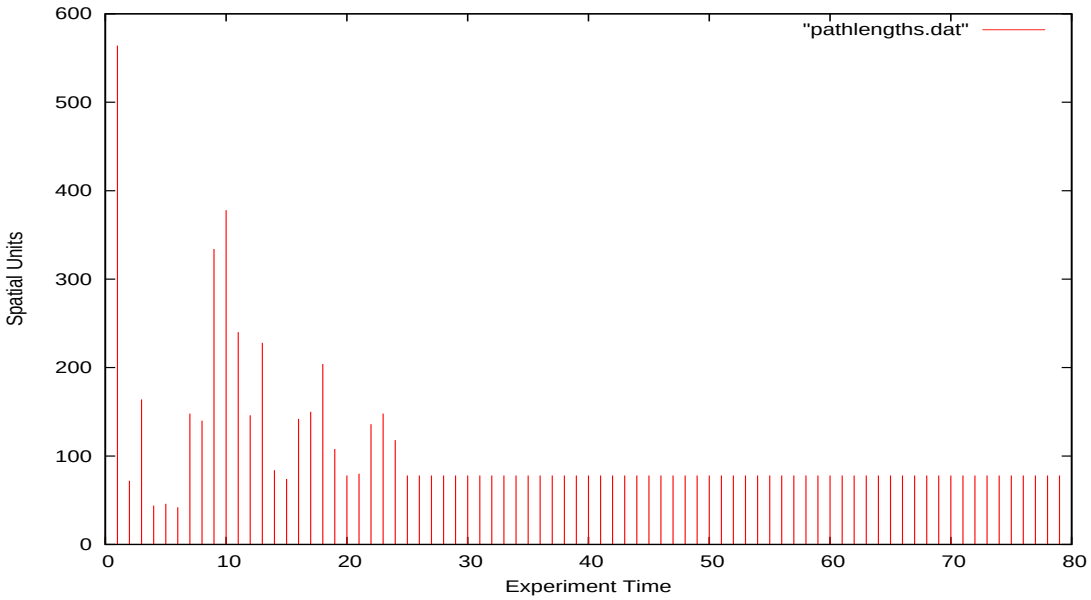

Figure F.41: Pathlengths from SW release point.

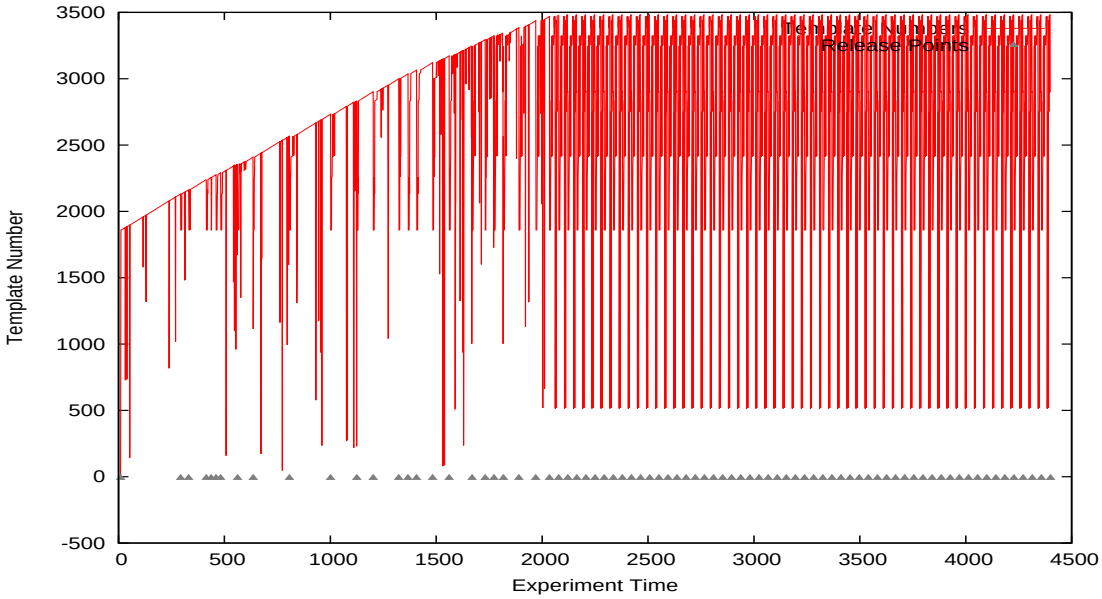

Figure F.42: Distal template formation from SW release point.

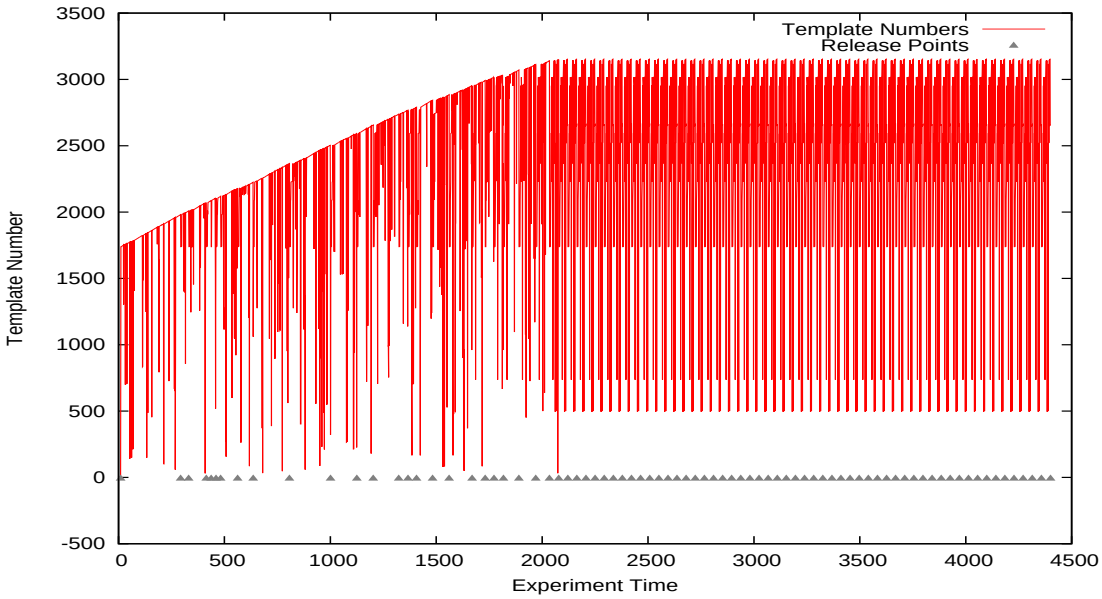

Figure F.43: Proximal template formation from SW release point.

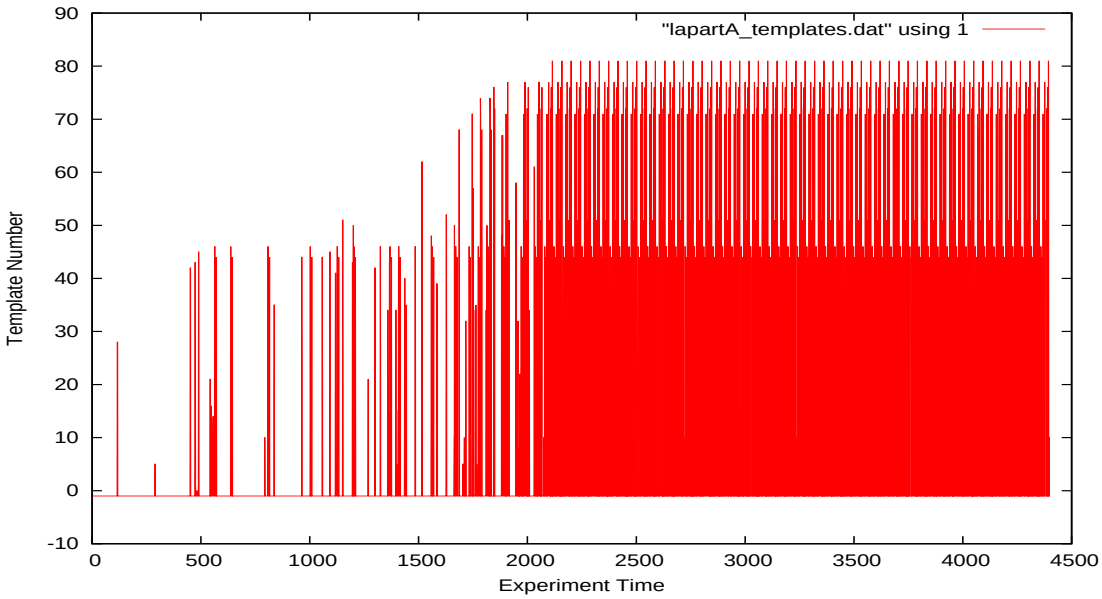

Figure F.44: LAPART A template formation from SW release point.

F.2.8 West then North release

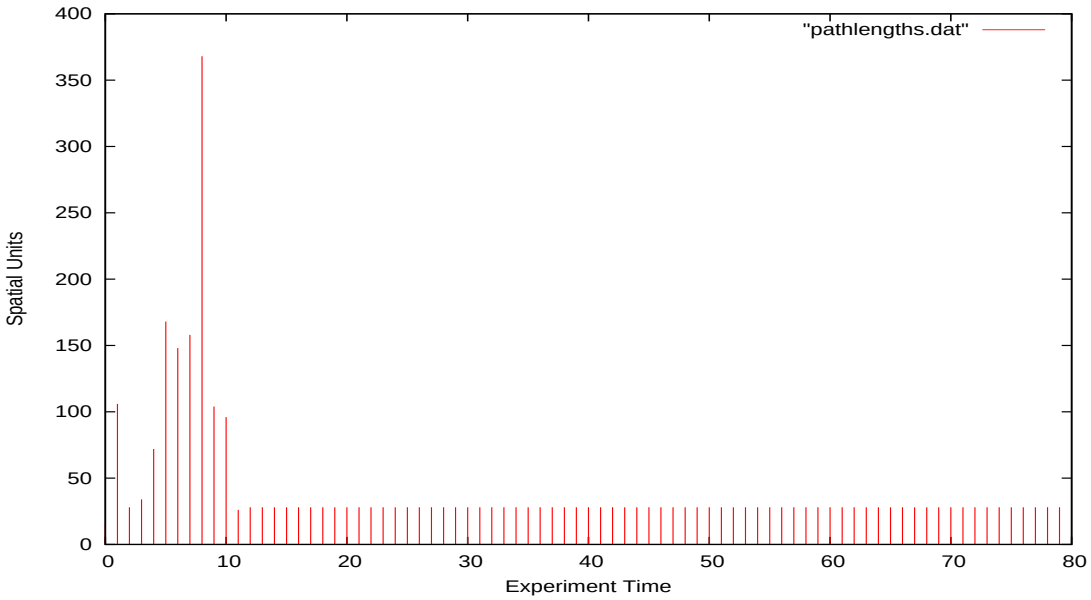

Figure F.45: Pathlengths from WN release point.

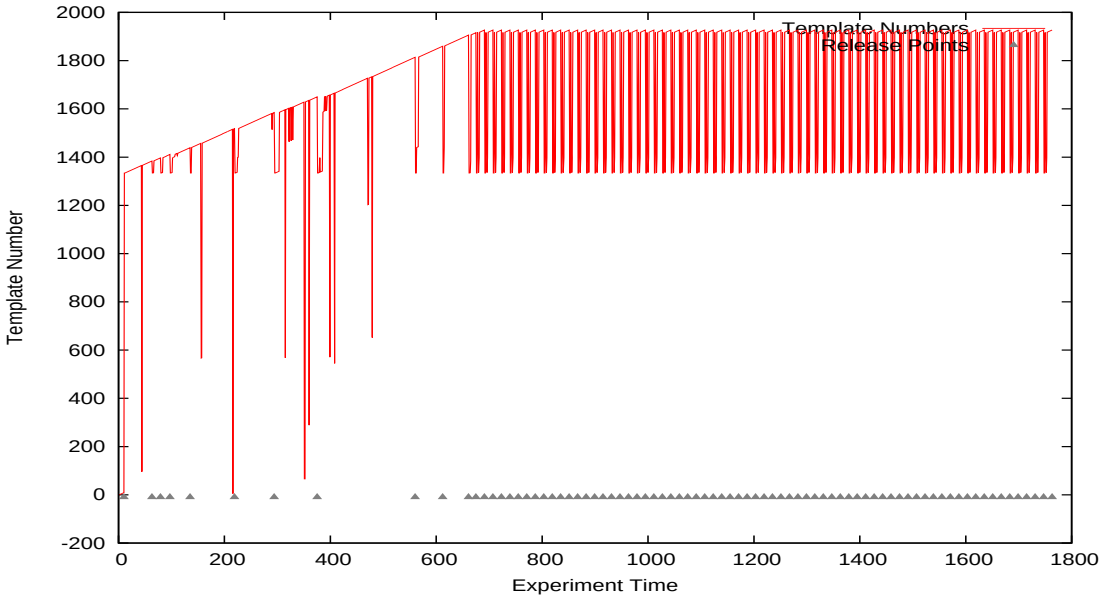

Figure F.46: Distal template formation from WN release point.

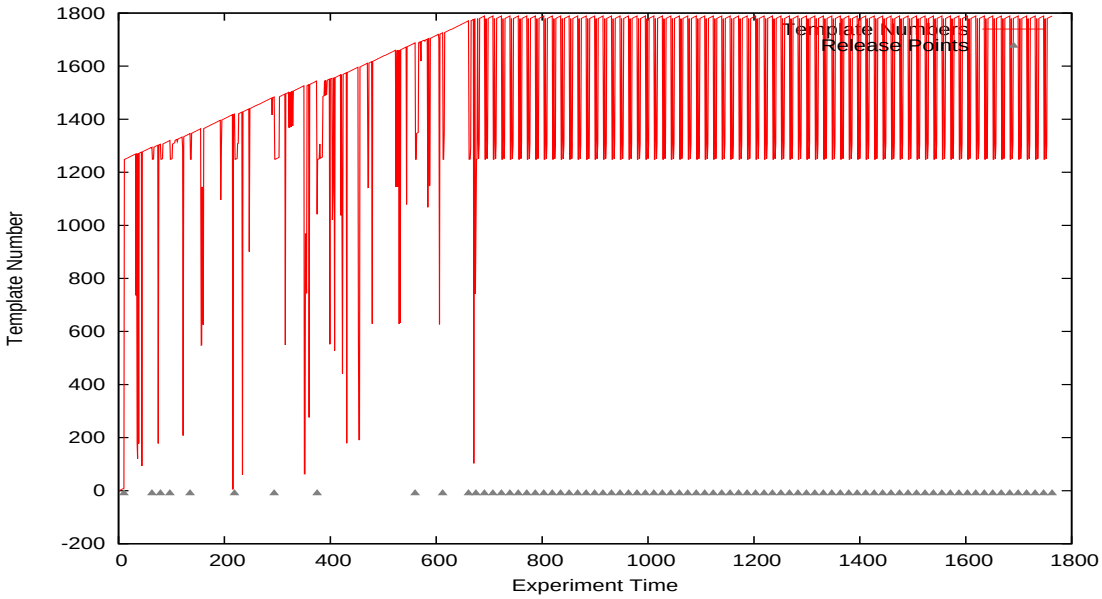

Figure F.47: Proximal template formation from WN release point.

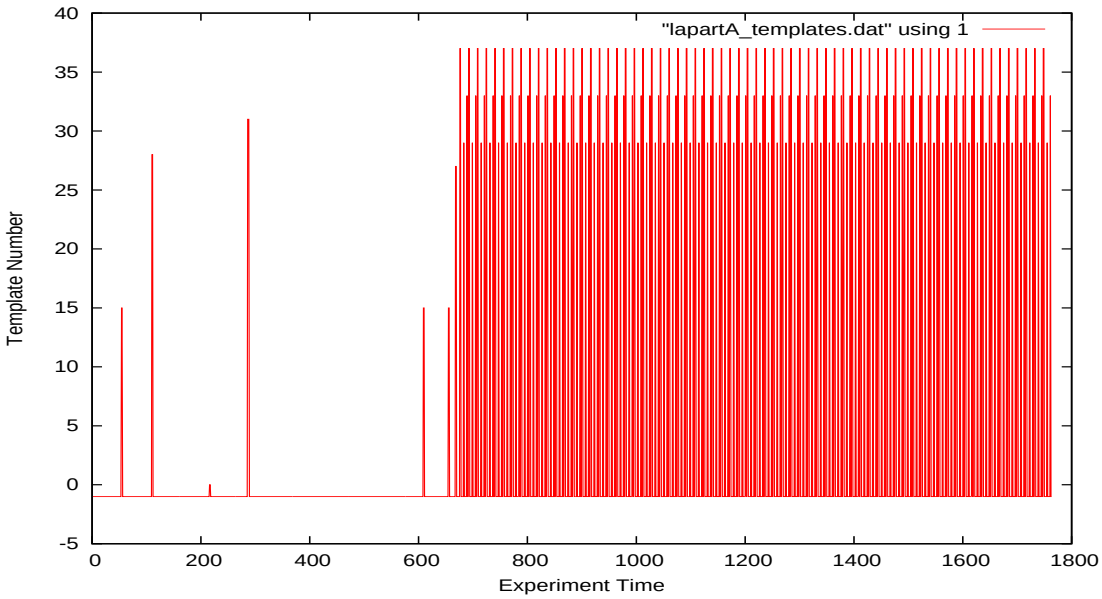

Figure F.48: LAPART A template formation from WN release point.

F.3 Block releases

These graphs depict data from the same experiments as previous graphs except pathlengths are summed over blocks. Recall that a block is a set of four realeases, one from each of the four release points.

F.3.1 Fixed Platform

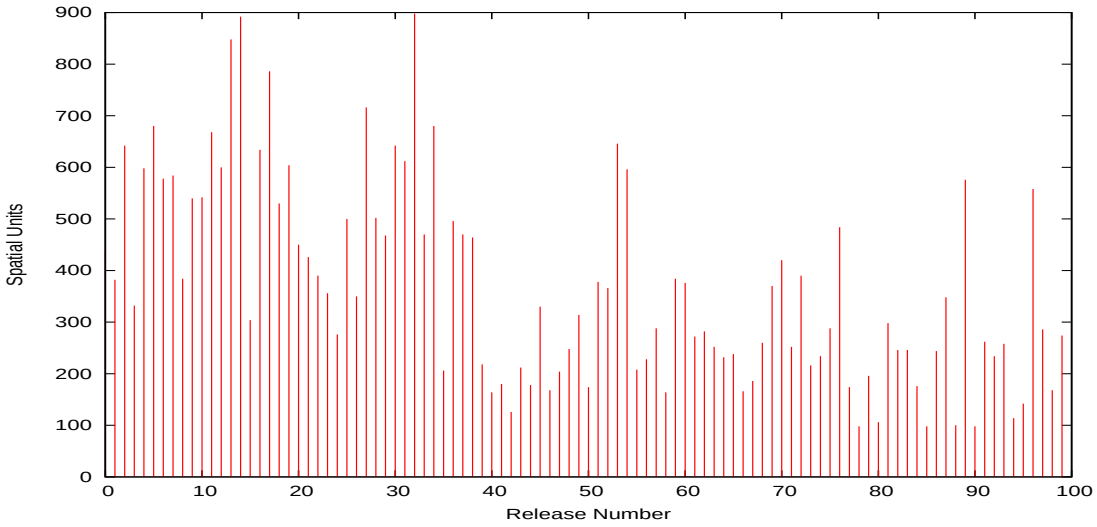

Figure F.49: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 0.

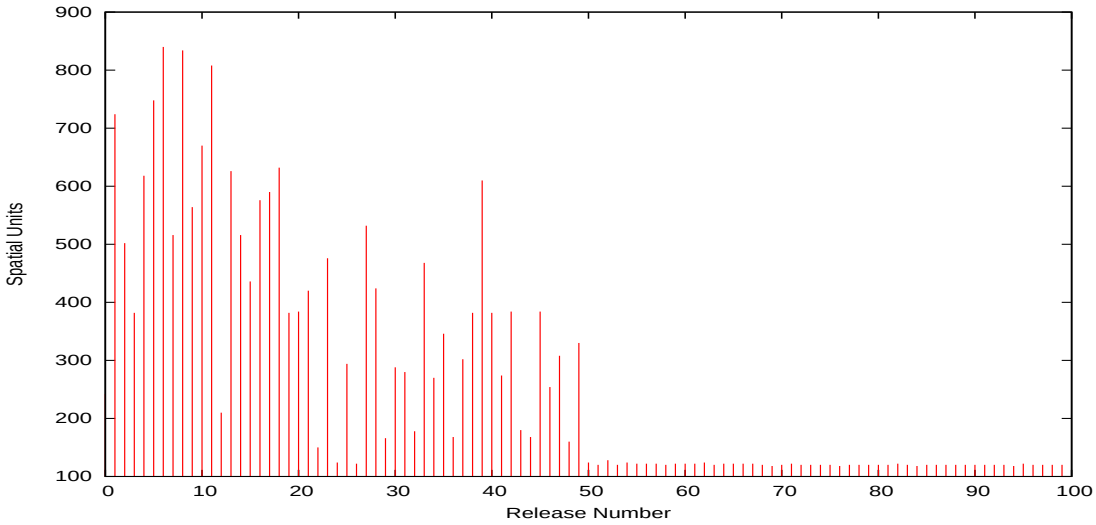

Figure F.50: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 1.

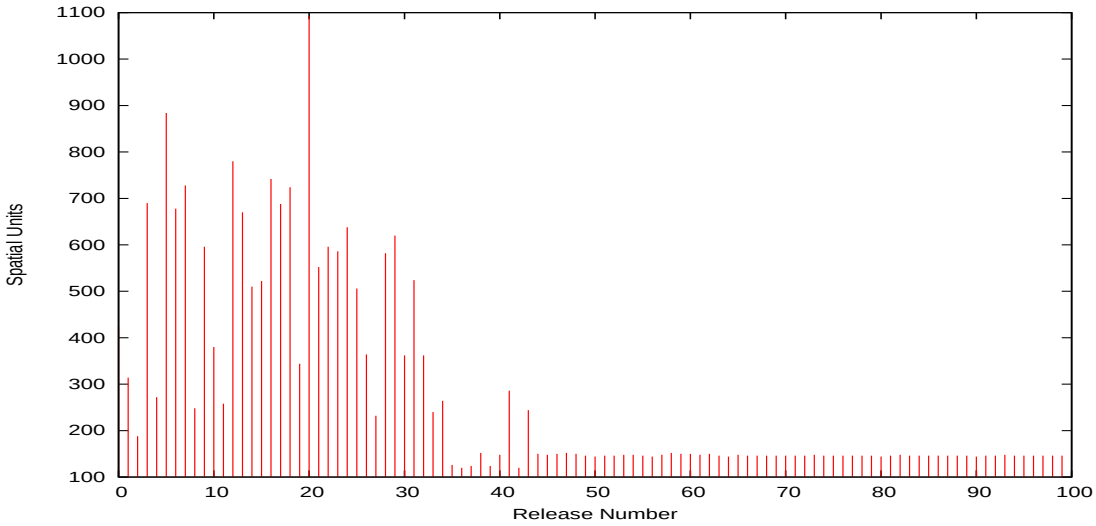

Figure F.51: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 2.

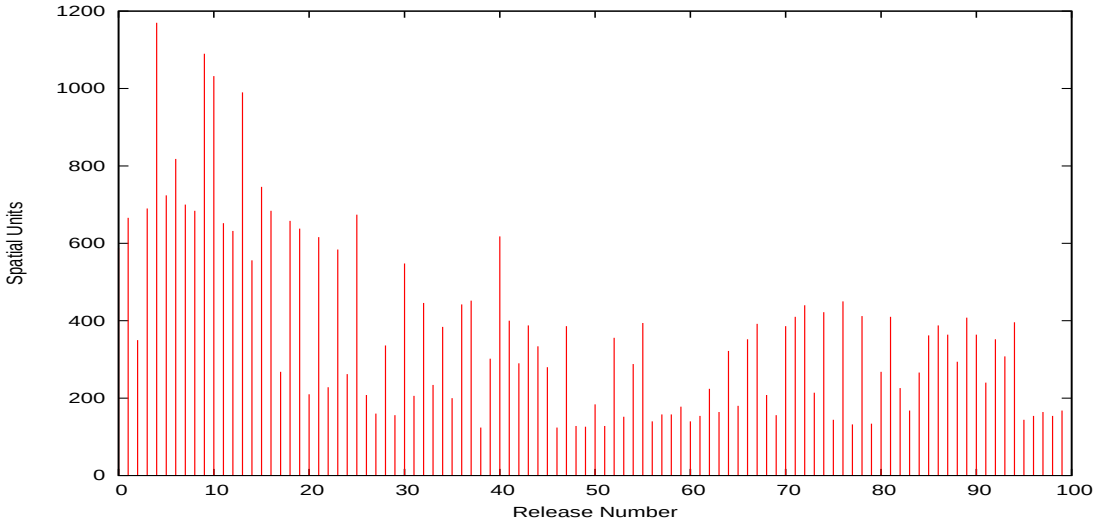

Figure F.52: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 3.

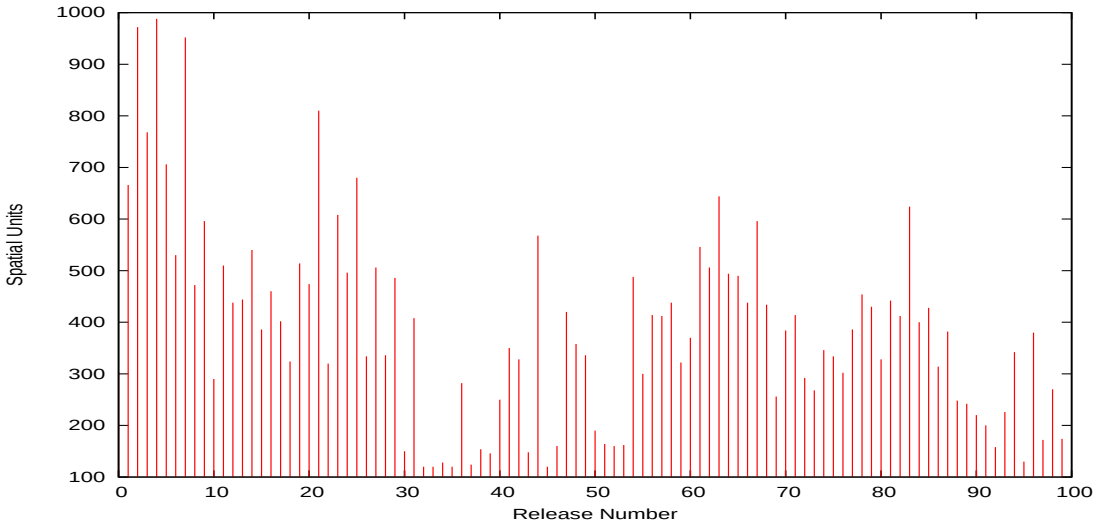

Figure F.53: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 4.

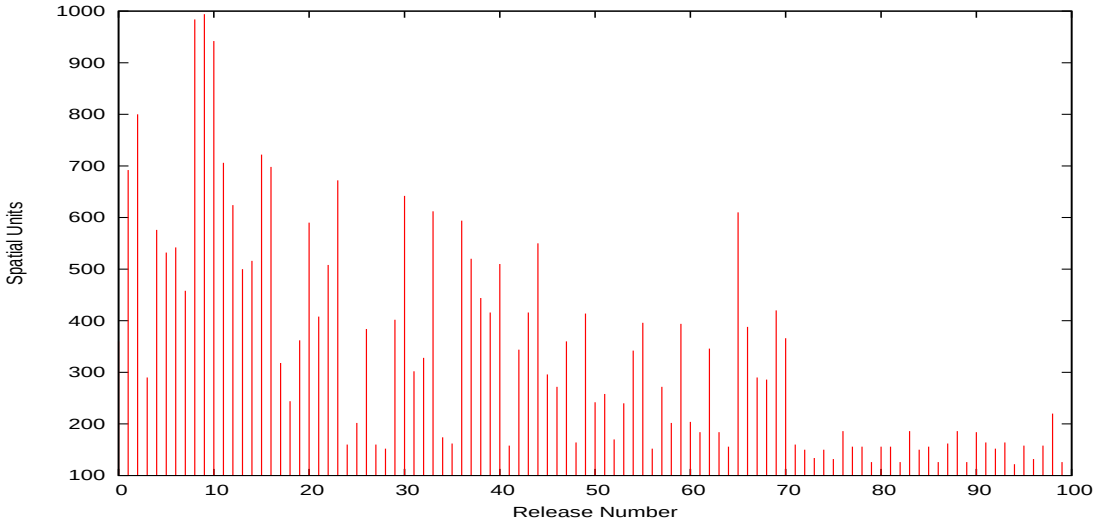

Figure F.54: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 5.

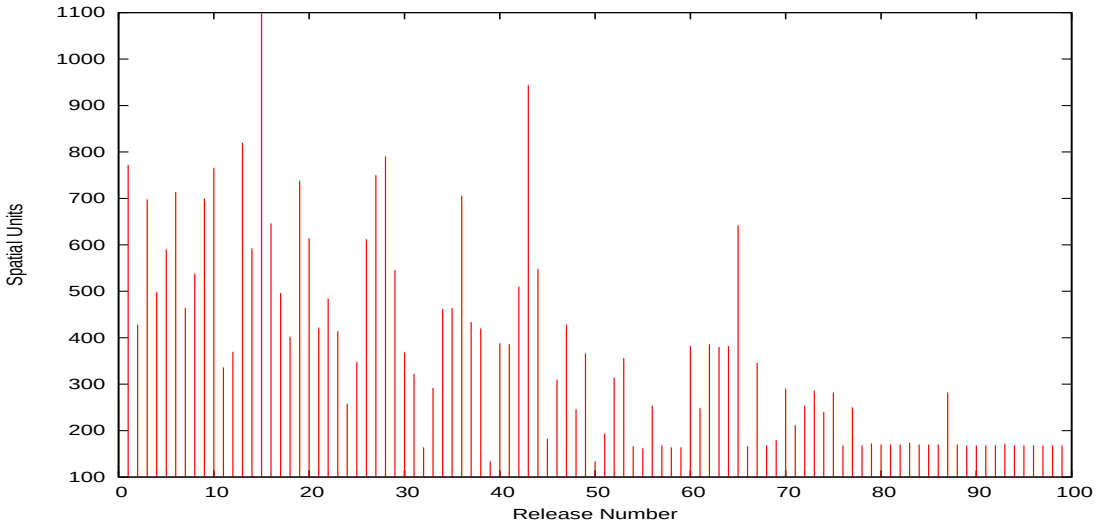

Figure F.55: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 6.

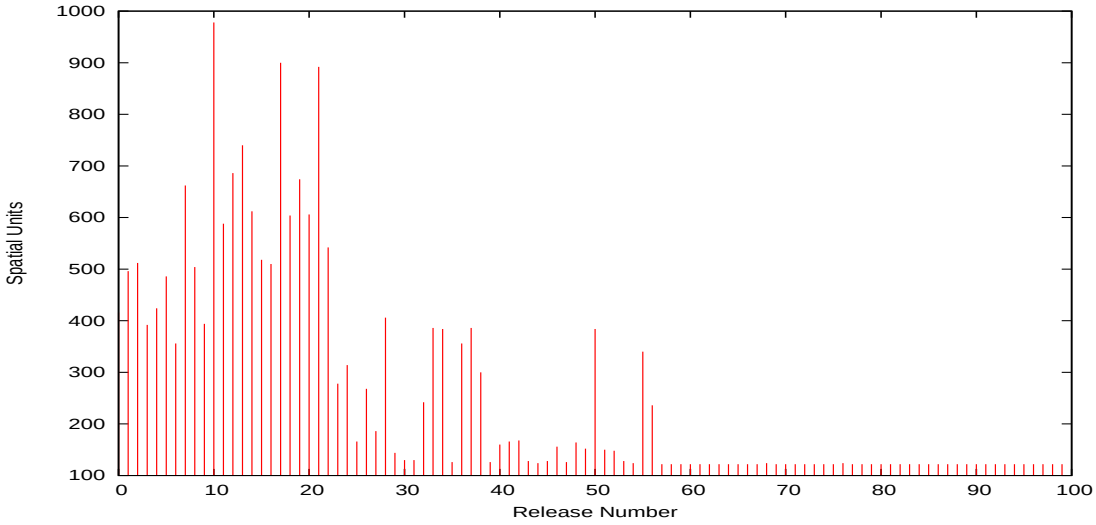

Figure F.56: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 7.

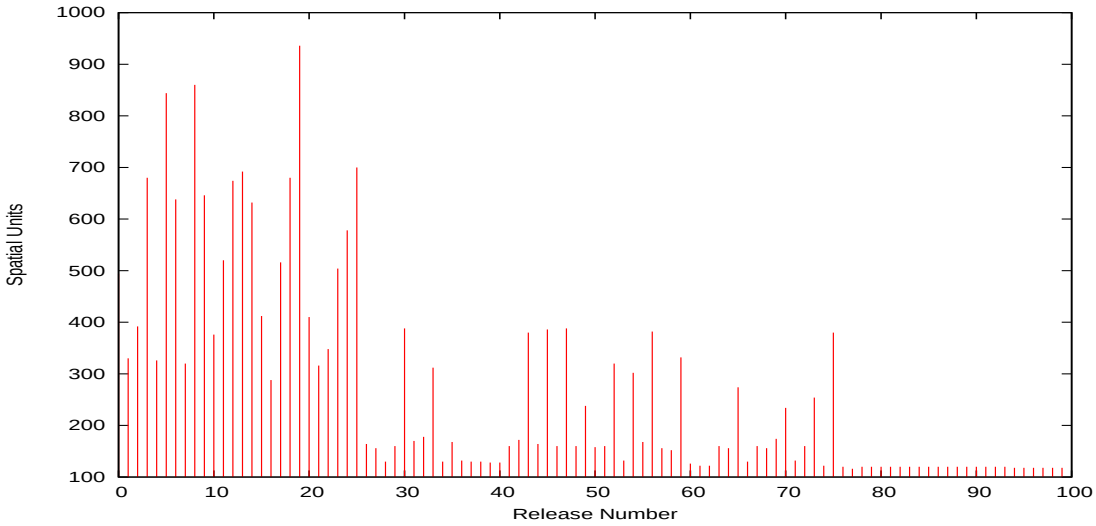

Figure F.57: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 8.

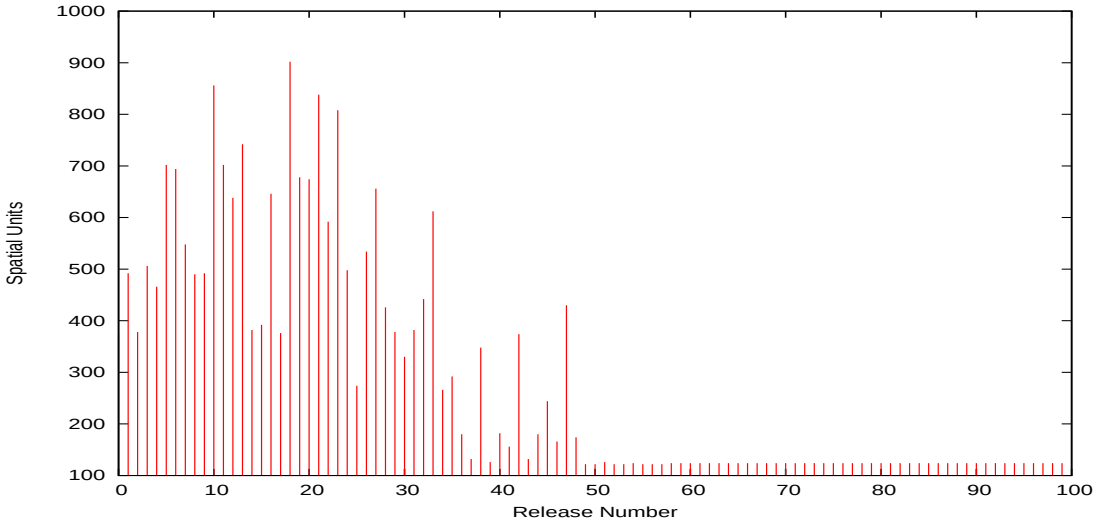

Figure F.58: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 9.

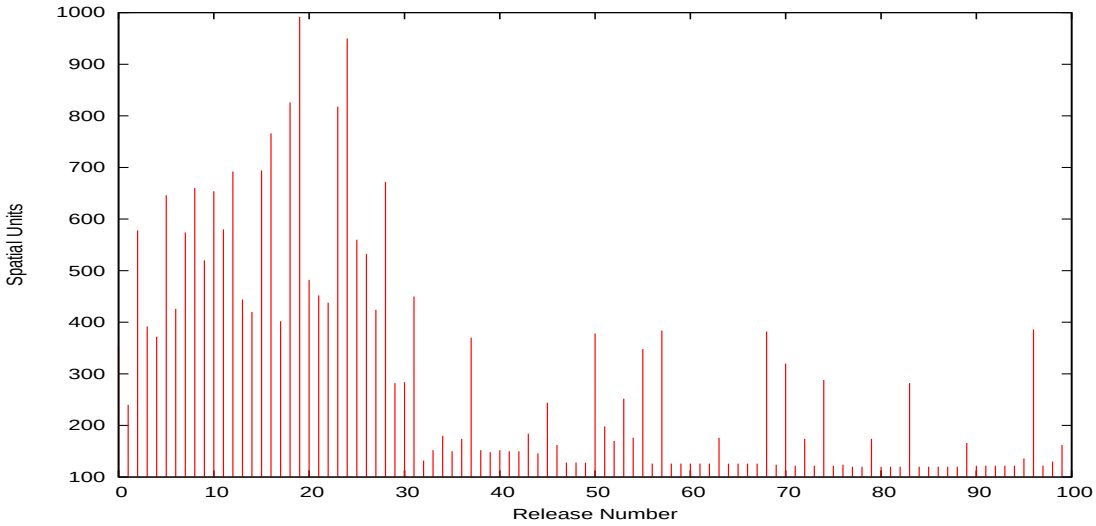

Figure F.59: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 10.

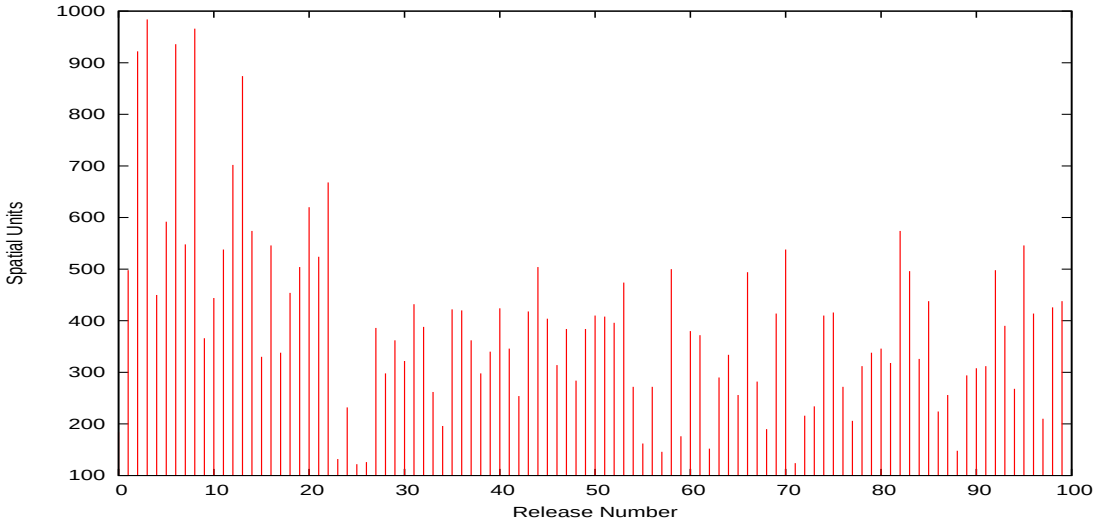

Figure F.60: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 11.

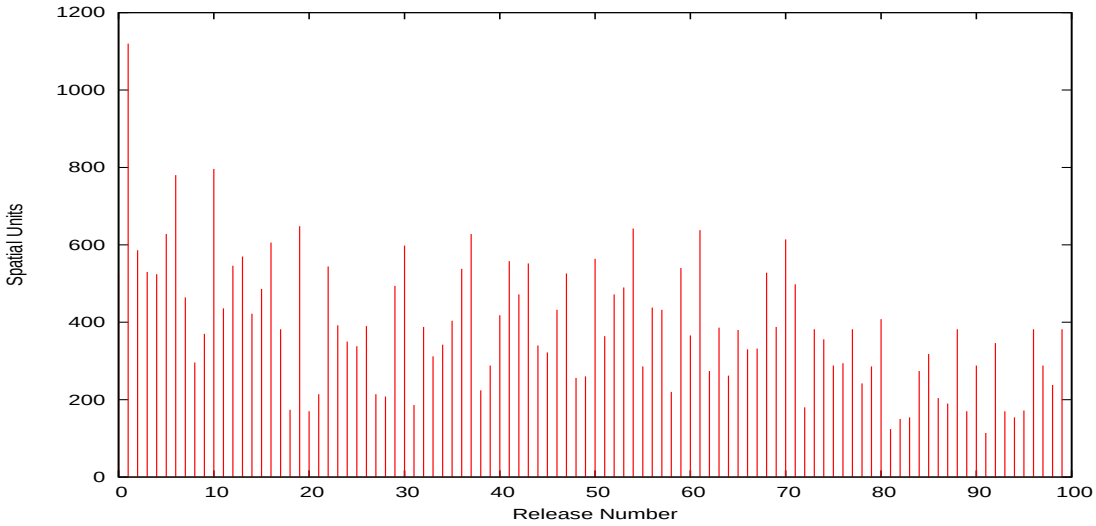

Figure F.61: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 12.

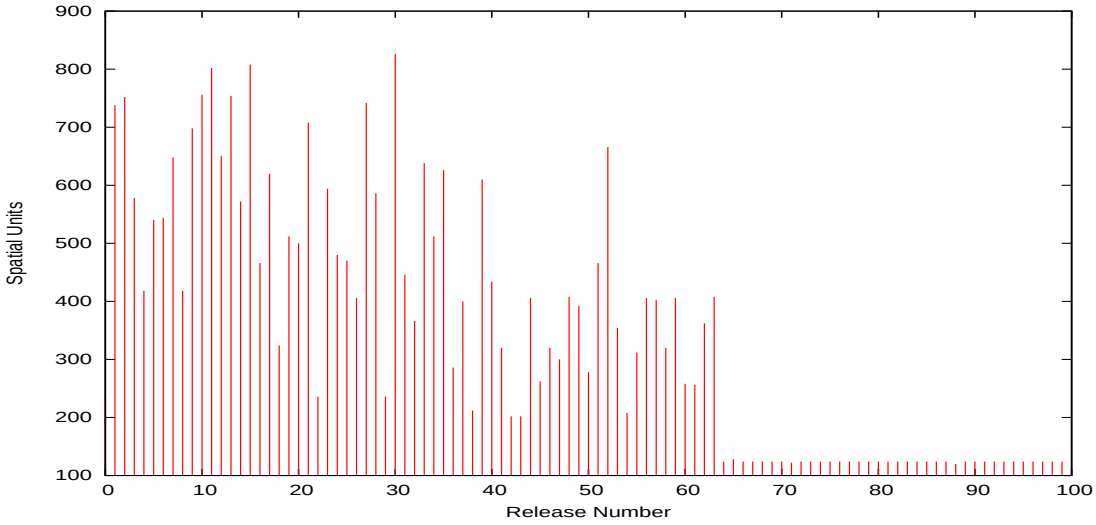

Figure F.62: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 13.

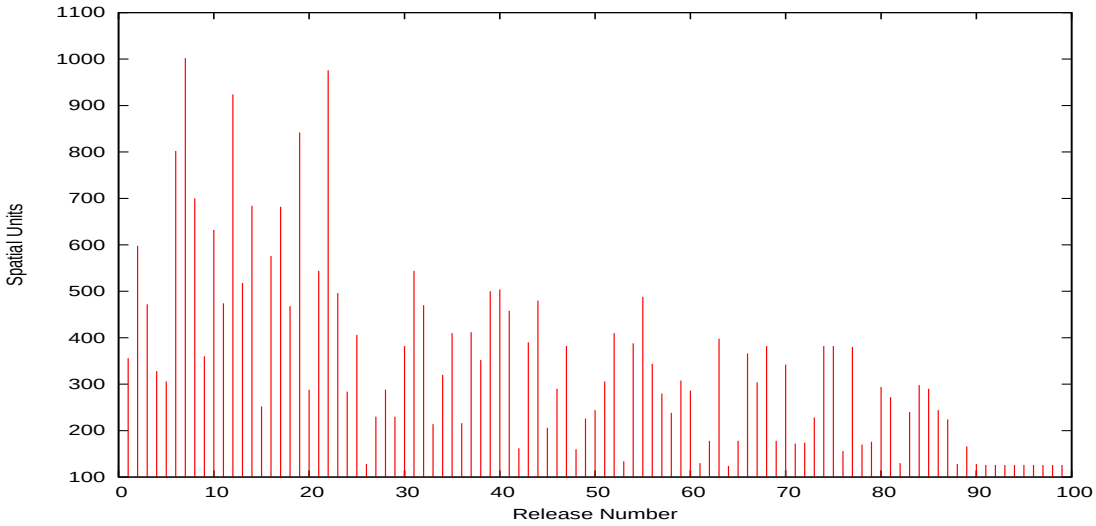

Figure F.63: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 14.

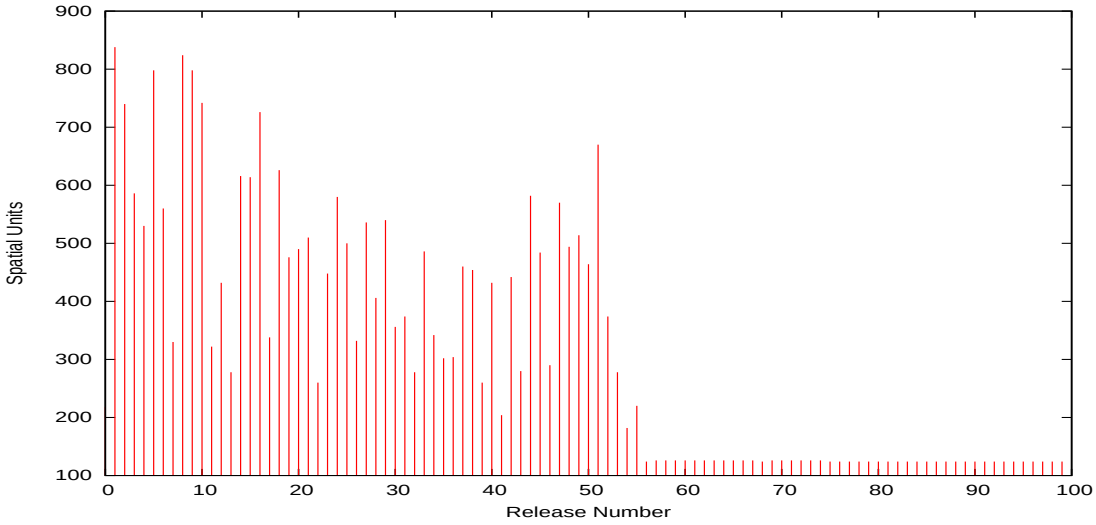

Figure F.64: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 15.

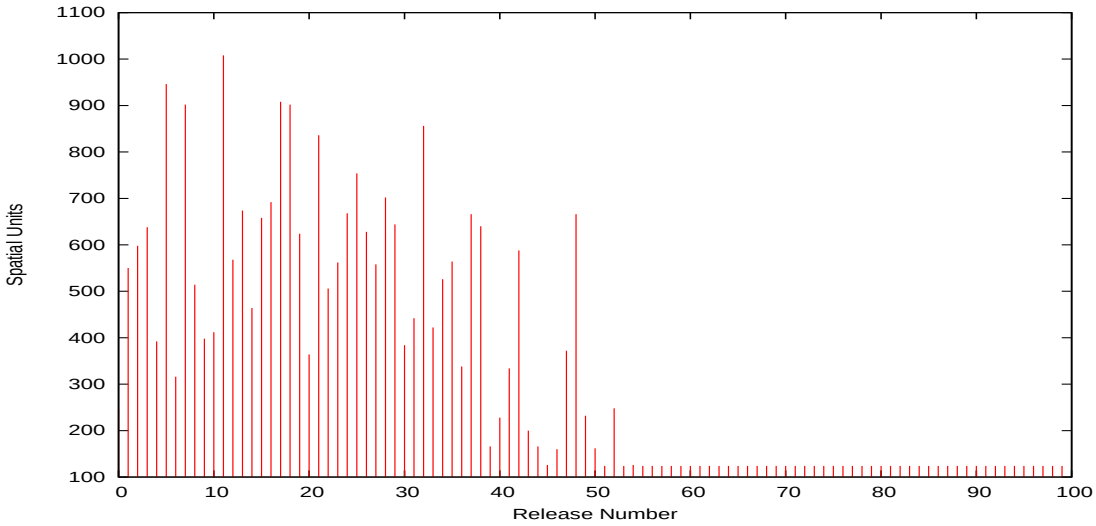

Figure F.65: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 16.

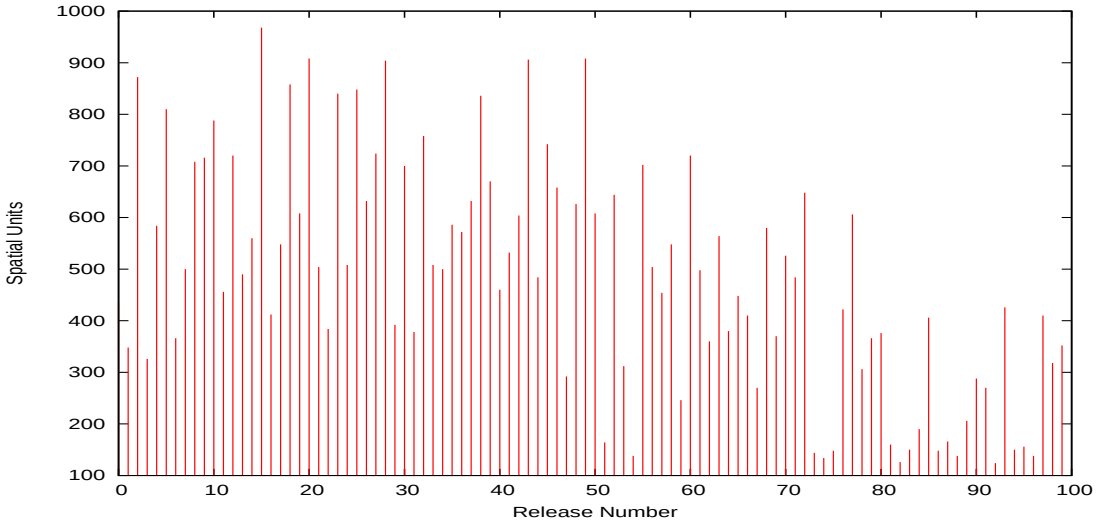

Figure F.66: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 17.

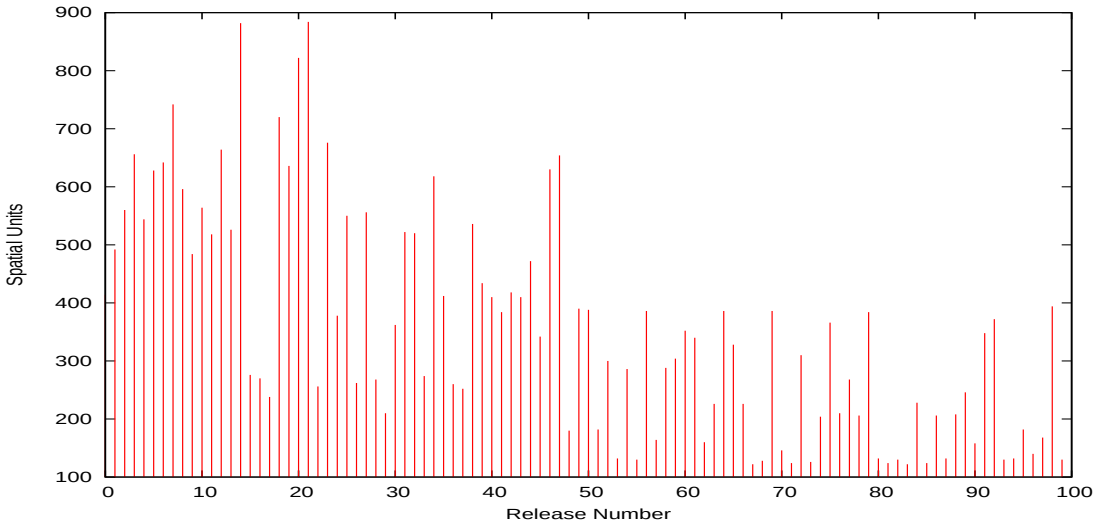

Figure F.67: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 18.

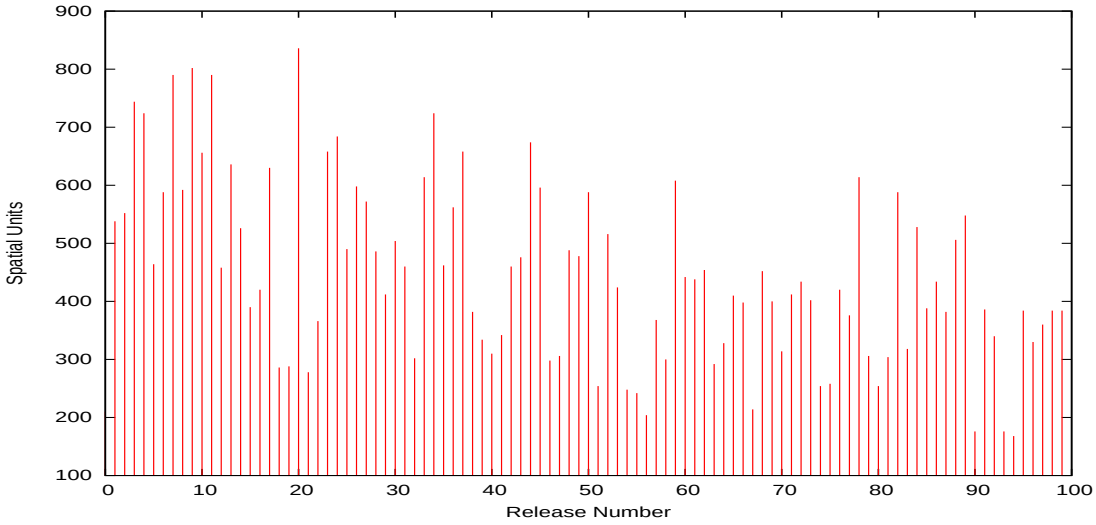

Figure F.68: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 19.

F.3.2 Random Platform Location

The purpose of the random platform location experiments are to serve as a control where no learning to a particular location should occur. At each release, the platform is found at a new, randomly selected, location.

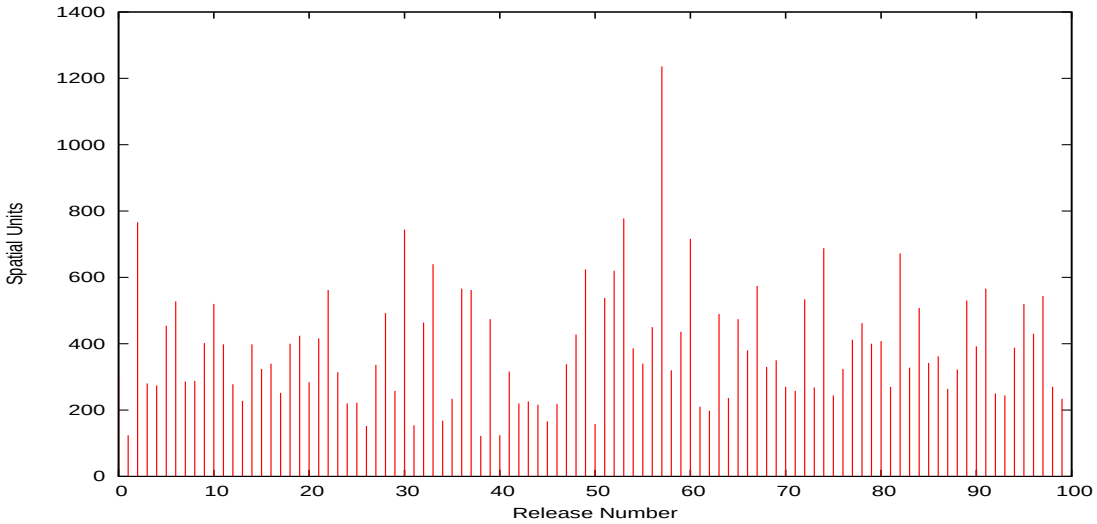

Figure F.69: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 0. Platform location changes for each release.

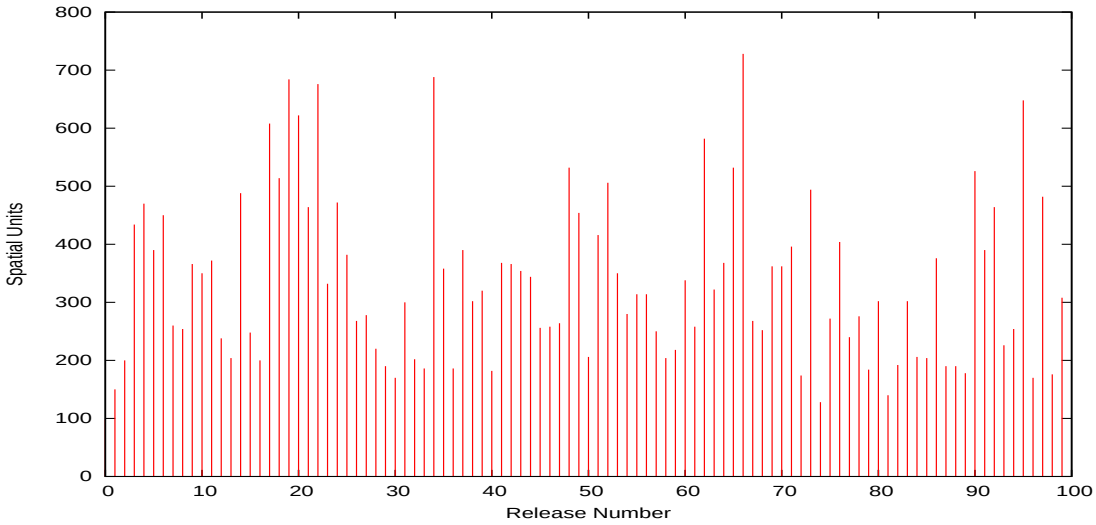

Figure F.70: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 1. Platform location changes for each release.

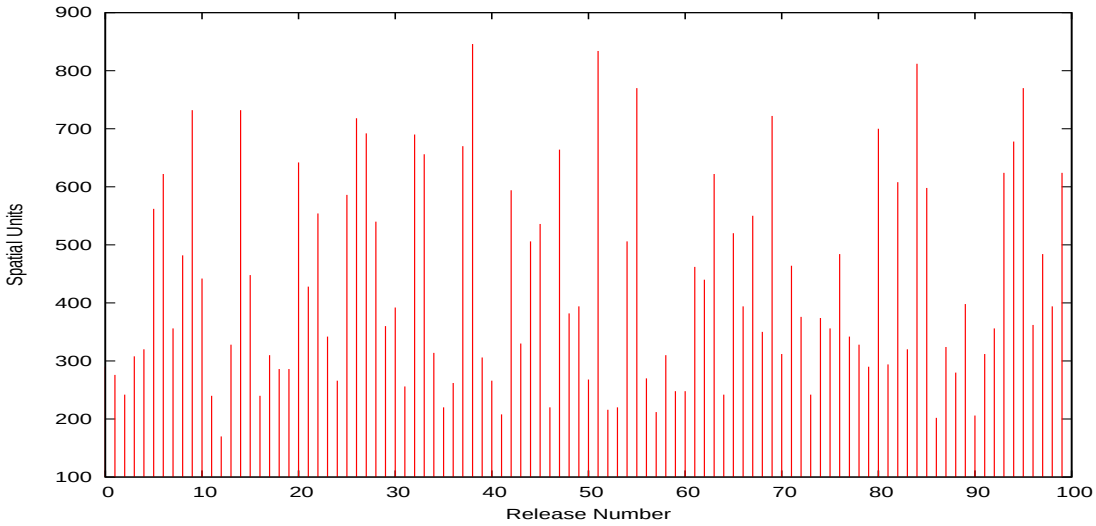

Figure F.71: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 2. Platform location changes for each release.

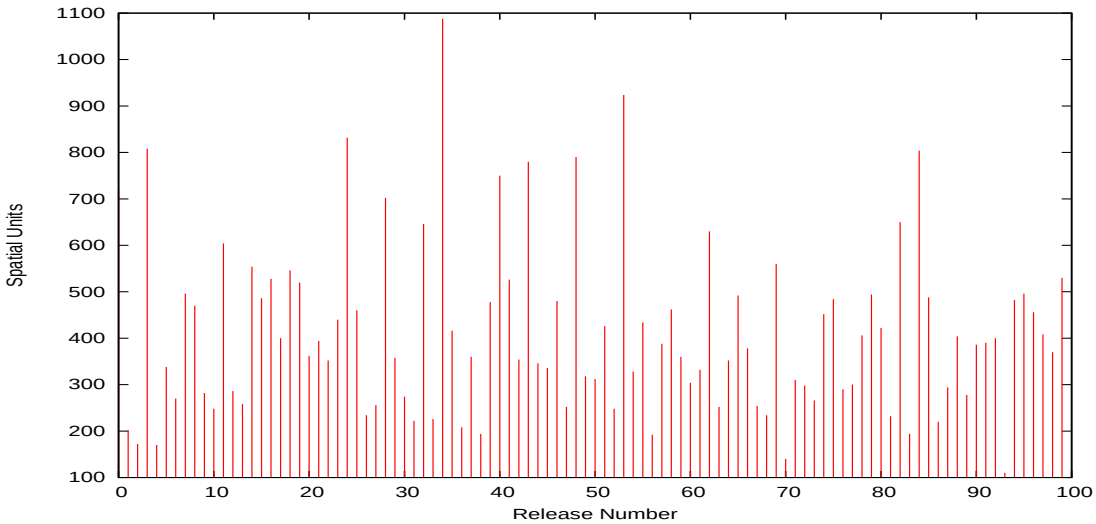

Figure F.72: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 3. Platform location changes for each release.

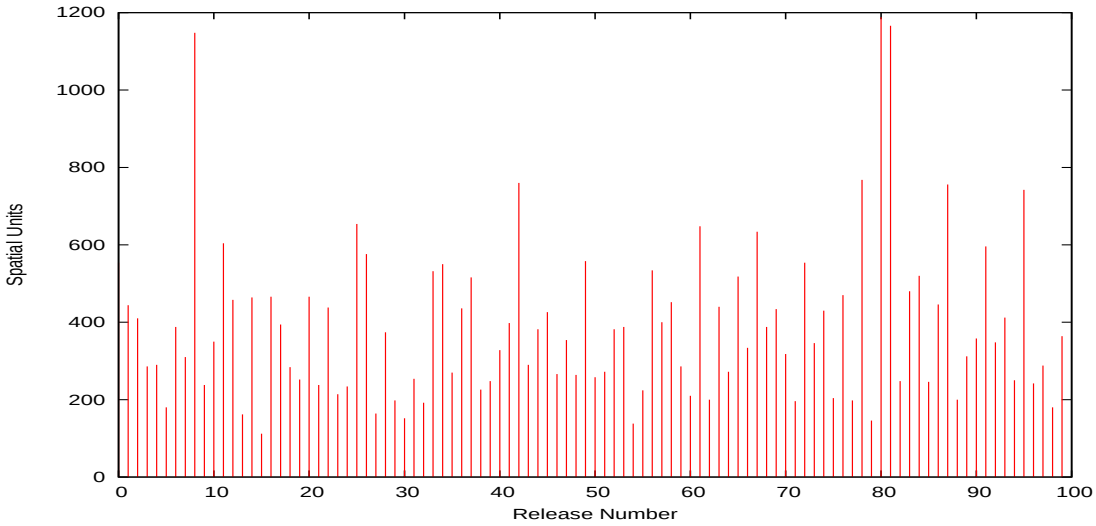

Figure F.73: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 4. Platform location changes for each release.

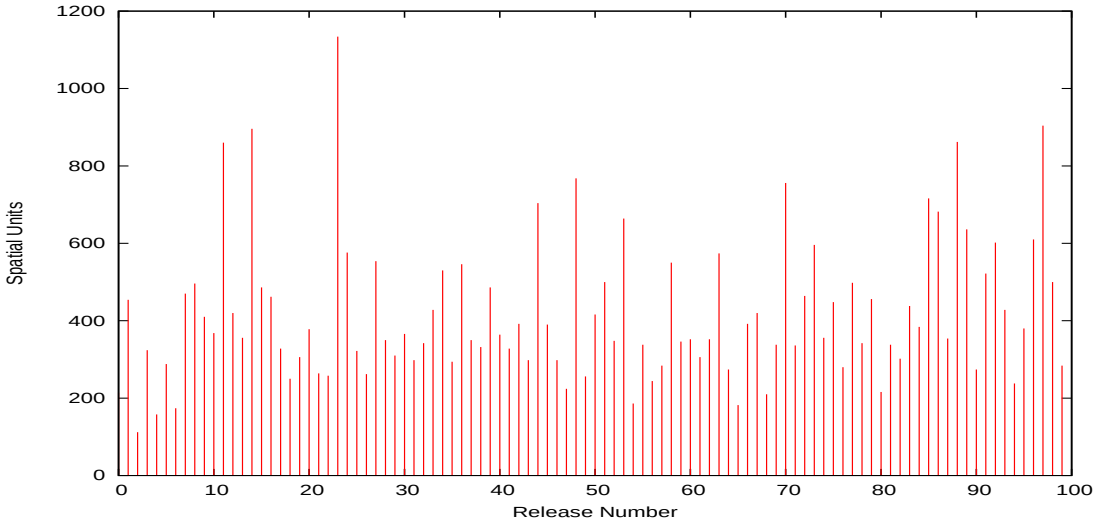

Figure F.74: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 5. Platform location changes for each release.

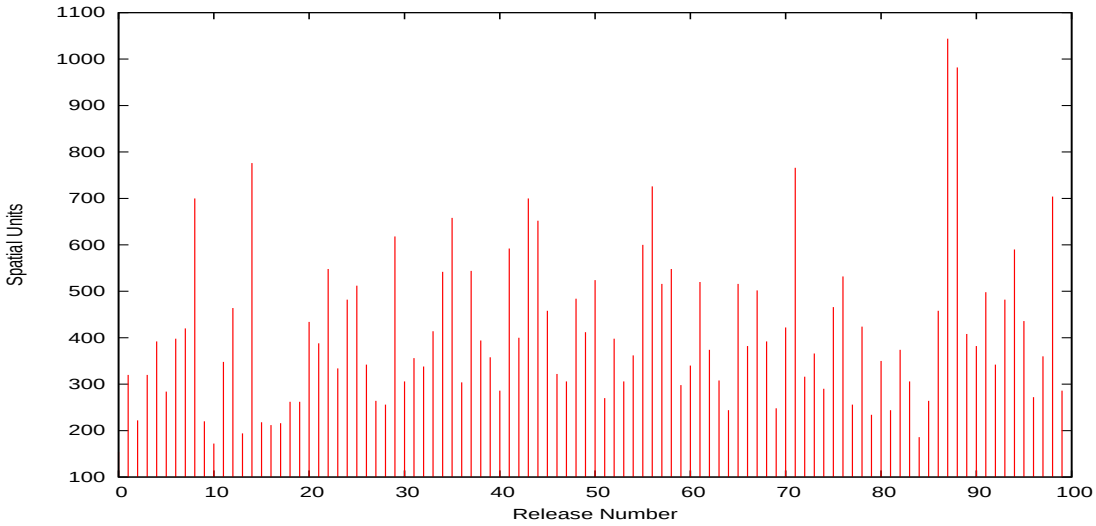

Figure F.75: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 6. Platform location changes for each release.

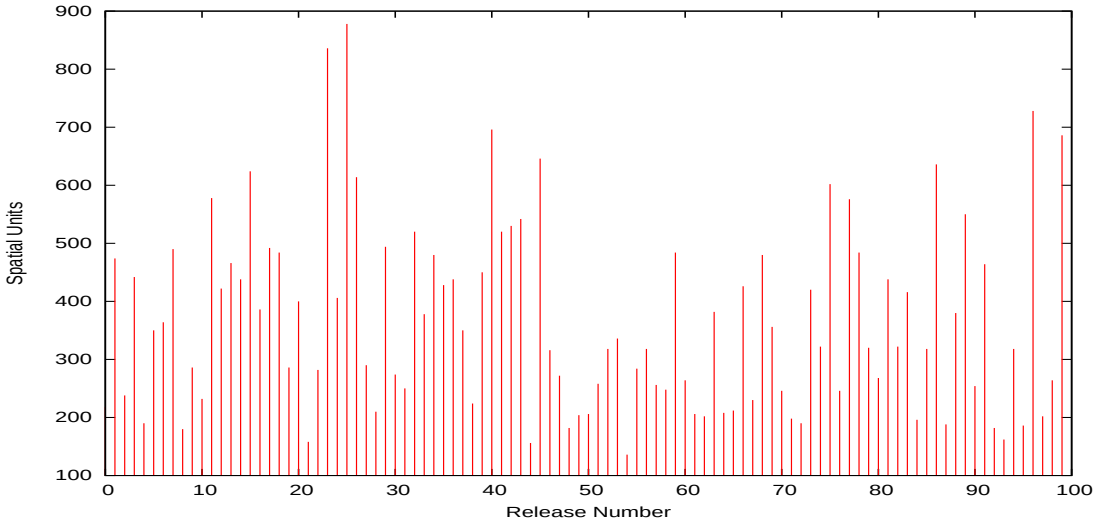

Figure F.76: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 7. Platform location changes for each release.

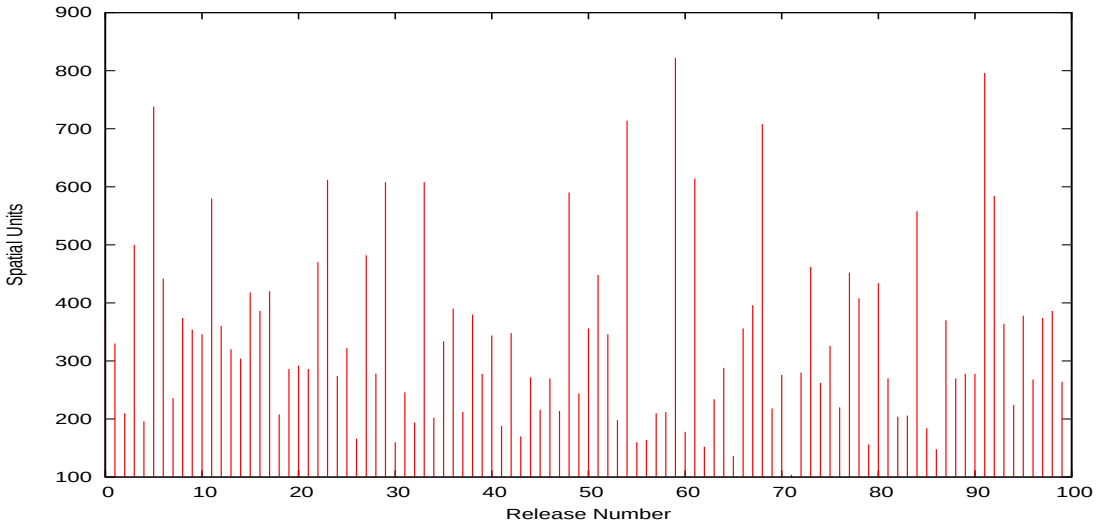

Figure F.77: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 8. Platform location changes for each release.

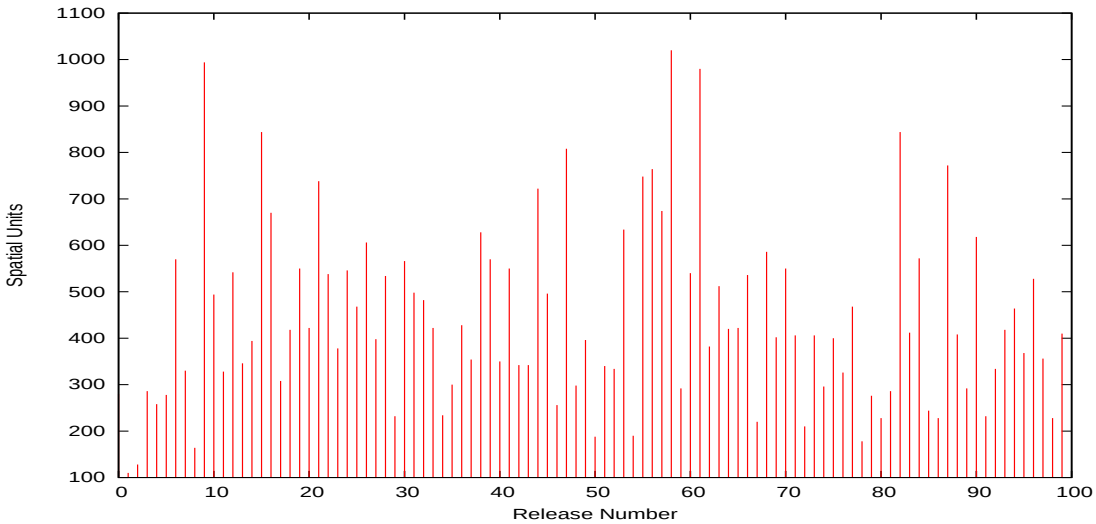

Figure F.78: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 9. Platform location changes for each release.

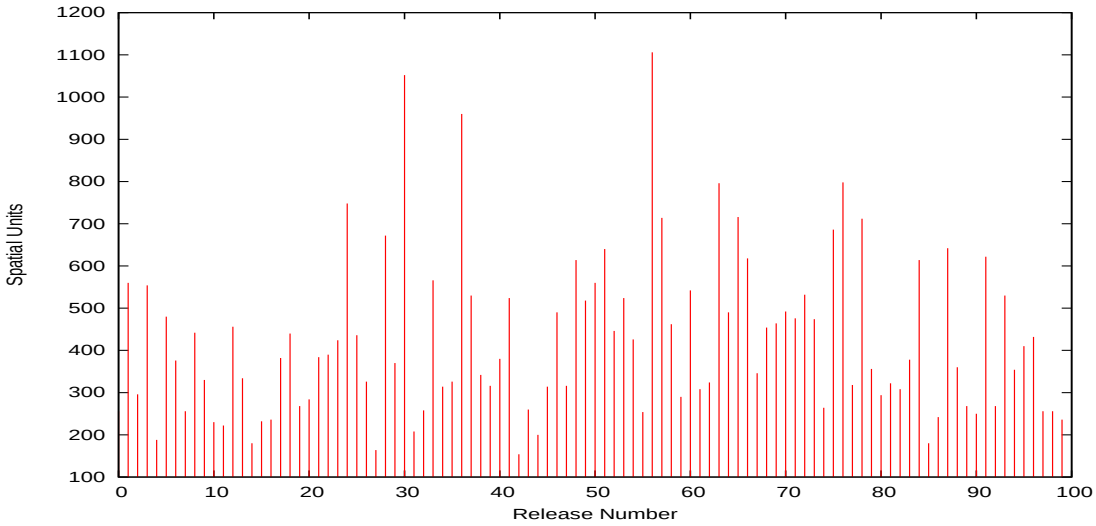

Figure F.79: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 10. Platform location changes for each release.

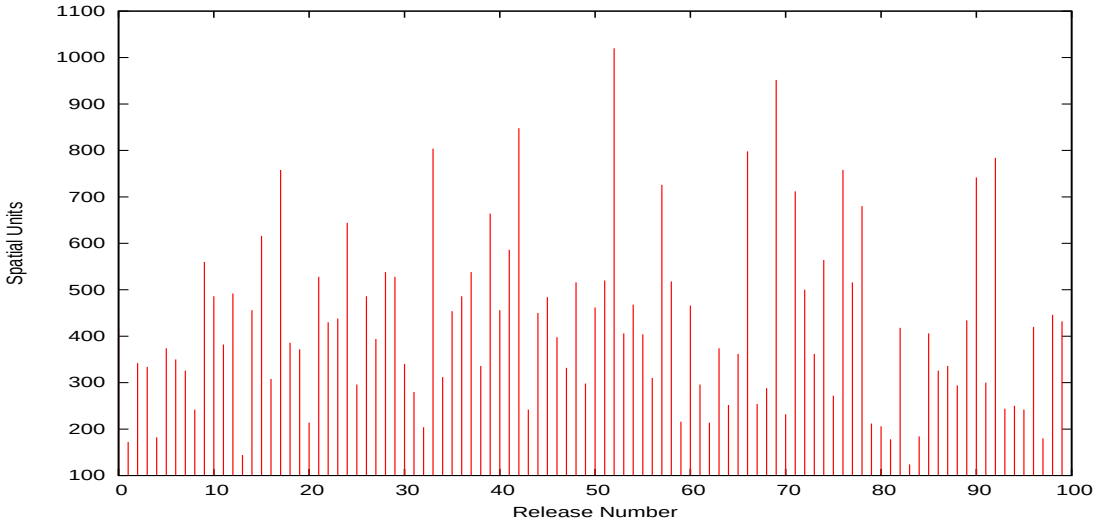

Figure F.80: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 11. Platform location changes for each release.

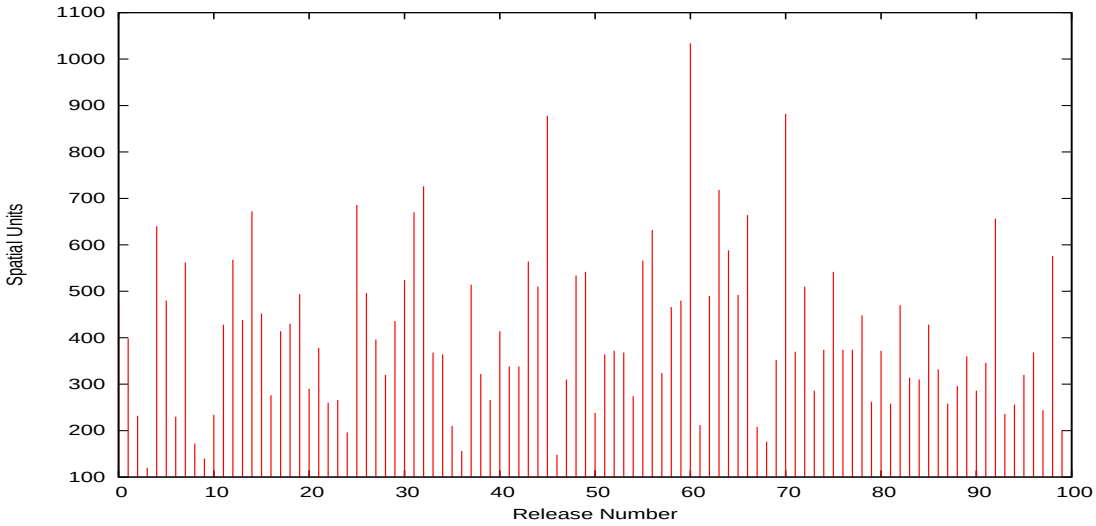

Figure F.81: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 12. Platform location changes for each release.

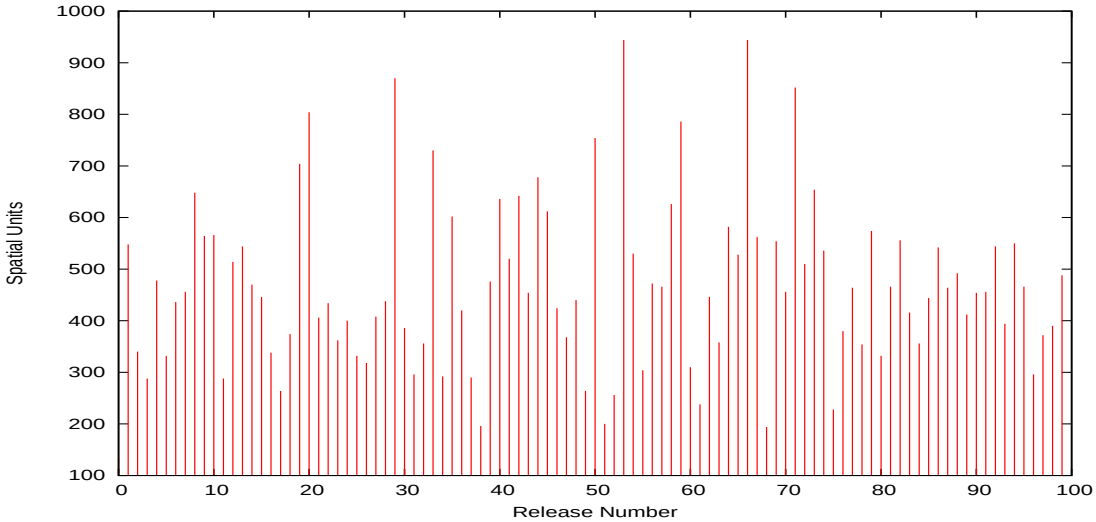

Figure F.82: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 13. Platform location changes for each release.

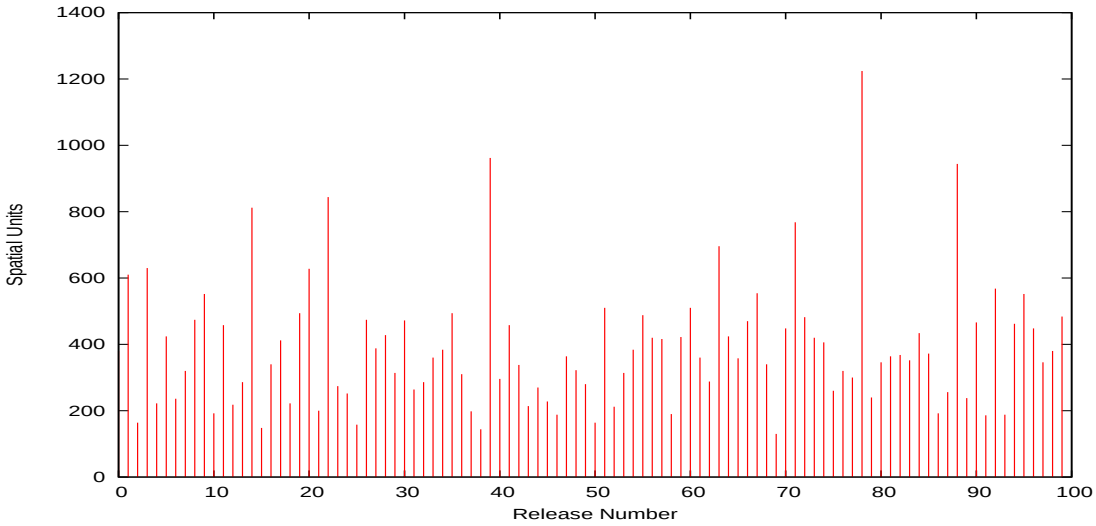

Figure F.83: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 14. Platform location changes for each release.

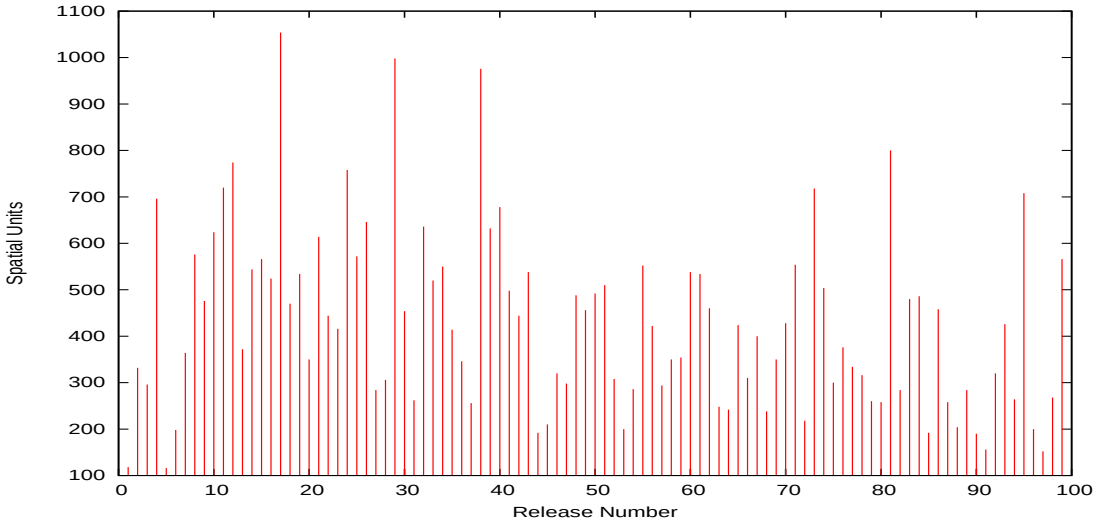

Figure F.84: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 15. Platform location changes for each release.

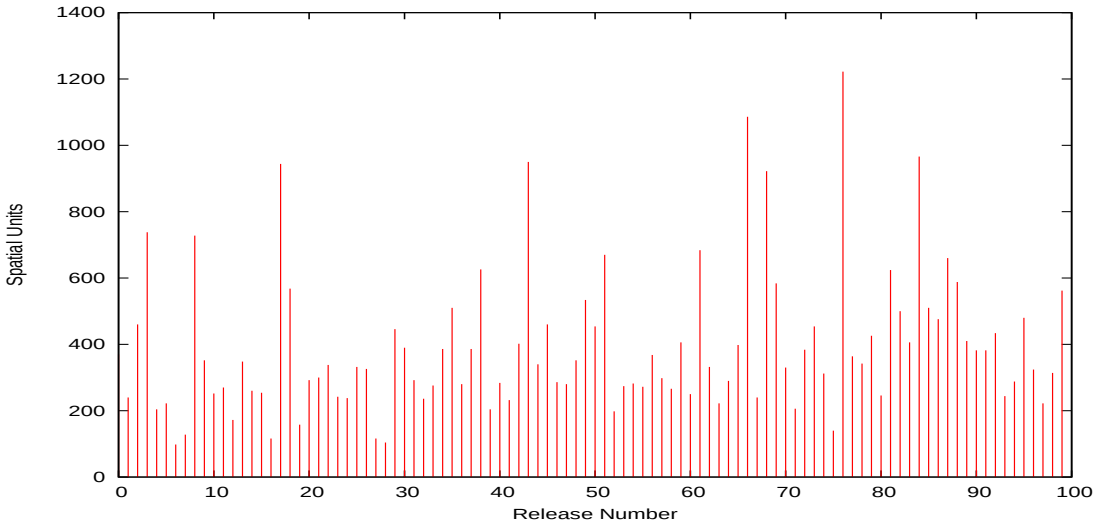

Figure F.85: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 16. Platform location changes for each release.

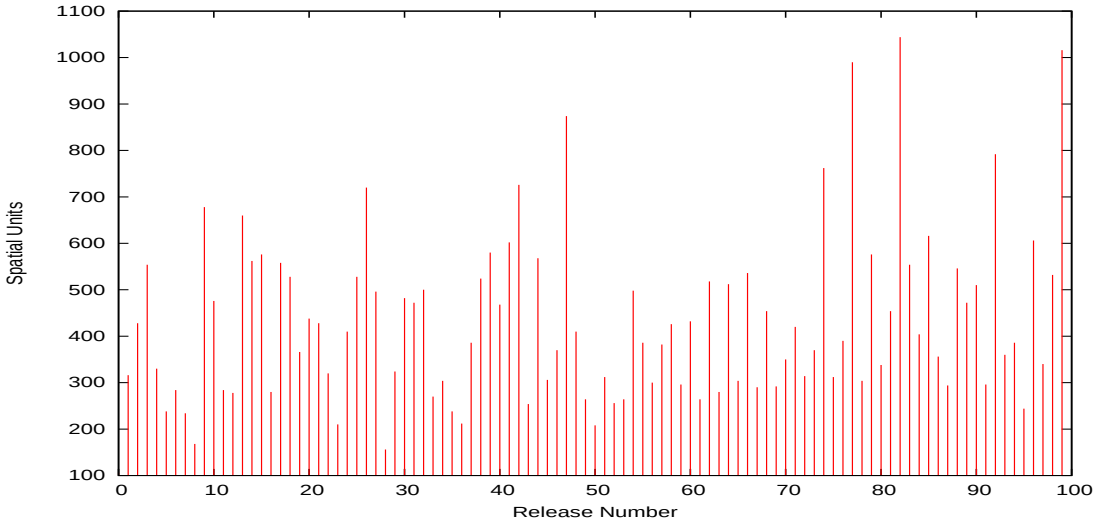

Figure F.86: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 17. Platform location changes for each release.

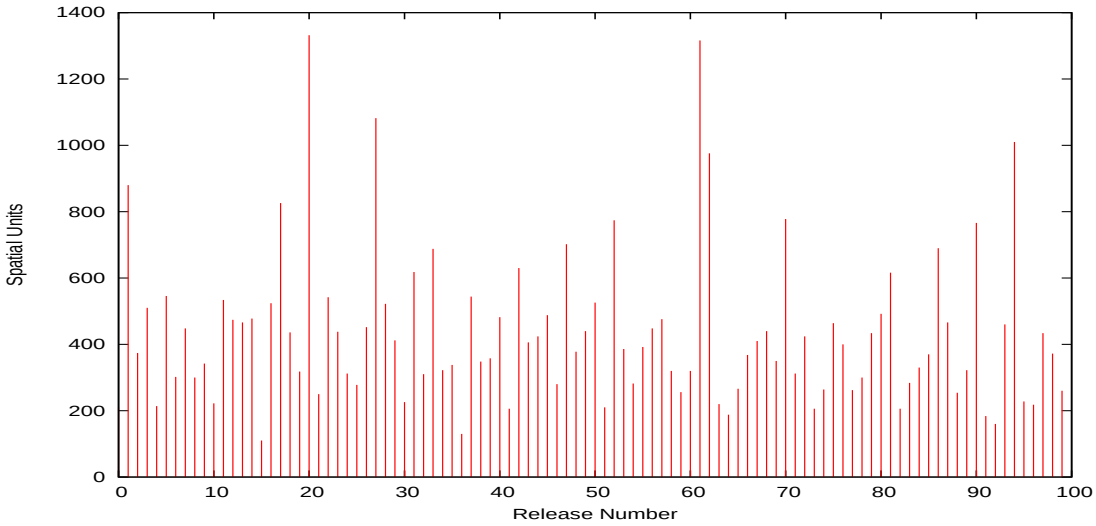

Figure F.87: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 18. Platform location changes for each release.

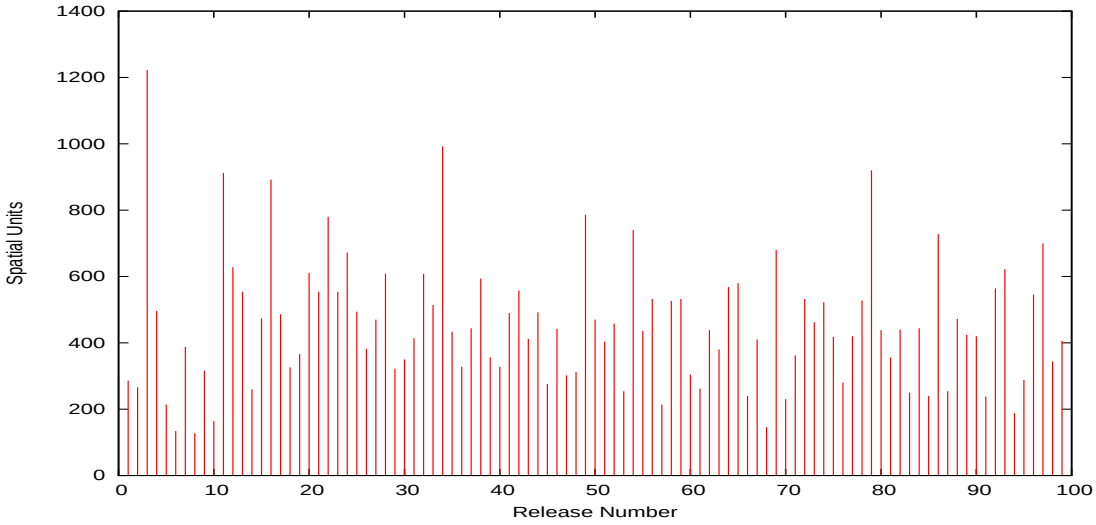

Figure F.88: Sum pathlengths by block of releases from random permutation of N, S, E, W for rat 19. Platform location changes for each release.

References

[1] J. L. McKinstry, A. K. Seth, G. M. Edelman, and J. L. Krichmar, *2008 special issue: Embodied models of delayed neural responses: Spatiotemporal categorization and predictive motor control in brain based devices*, Neural Netw. **21** (2008), 553–561.

[2] M. J. Healy and T. P. Caudell, *Acquiring rule sets as a product of learning in a logical neural architecture*, IEEE Transactions on Neural Networks **8** (1997), 461–474.

[3] R. W. Williams and K. Herrup, *The control of neuron number*, Annual Review of Neuroscience **11** (1988), 423453.

[4] S. E. Taylor, M. J. Healy, and T. P. Caudell, *Categorical mapping from ontology to neural network: Initial studies of simple neural networks' concept capacity*, in Proceedings of the International Joint Conference on Neural Networks, Institute of Electrical and Electronic Engineers, Orlando, FL, USA, Aug. 12–17 2007, pp. 2020–2025.

[5] I. P. Pavlov, *Conditioned reflexes*, Oxford University Press, Oxford, 1927.

[6] C. L. Hull, *Principles of behavior*, Appleton-Century, New York, 1943.

[7] H. JC, D. JL, and B. DG, *Models of information processing in the basal ganglia*, The MIT Press, Cambridge, MA, 1995.

[8] W. Schultz, *Predictive reward signal of dopamine neurons*, Neurophysiology **80** (2007), 1–27.

[9] D. P and A. LF, *Theoretical neuroscience: computational and mathematical modeling of neural systems*, The MIT Press, Cambridge, MA, 2001.

[10] M. L. Minsky, *Steps toward artificial intelligence*, in Computers and thought, (F. EA and F. J, eds.), McGraw-Hill, New York, 1963, pp. 406–450.

[11] A. Barto, R. Sutton, and C. Anderson, *Neuronlike elements that can solve difficult learning control problems*, IEEE Trans Syst Man Cybern. **13** (1983), 835–846.

[12] H. JC, A. JL, and B. AG, *A model of how the basal ganglia generate and use neural signals that predict reinforcement*, in Models of information processing in the basal ganglia, (H. JC, D. JL, and B. DG, eds.), The MIT Press, Cambridge, MA, 1995, pp. 249–270.

[13] S. RS and B. AG, *Reinforcement learning: an introduction*, The MIT Press, Cambridge, MA, 1998.

[14] W. F and P. B, *Temporal sequence learning, prediction, and control: a review of different models and their relation to biological mechanisms*, Neural Comput. **15** (2005), 245–319.

[15] E. F. Keller and L. A. Segel, *Initiation of slime mold aggregation viewed as an instability*, Journal of Theoretical Biology **26** (1970), 399–415.

[16] H. C. Berg and D. A. Brown, *Chemotaxis in Escherichia coli analysed by Three-dimensional Tracking*, Nature **239** (1972), 500–504.

[17] D. A. Nitz, W. J. Kargo, and J. Fleischer, *Dopamine signaling and the distal reward problem*, NeuroReport **18** (2007), 1833–1836.

[18] R. Kingsley, *Concise Text of Neuroscience*, Lippincott Williams & Wilkins, 2000.

[19] E. M. Izhikevich, *Solving the Distal Reward Problem through Linkage of STDP and Dopamine Signaling*, Cerebral Cortex **17** (2007), 2443–2452.

[20] ———, *Polychronization: computation with spikes*, Neural Comput. **18** (2006), 245–282.

[21] B. W. Connors and M. J. Gutnick, *Intrinsic firing patterns of diverse neocortical neurons*, Trends in Neurosciences **13** (1990), 99 – 104.

[22] A. V. Samsonovich and G. A. Ascoli, *A simple neural network model of the hippocampus suggesting its pathfinding role in episodic memory retrieval*, Learning and Memory **12** (2005), 193–208.

[23] J. O'Keefe and M. L. Recce, *Phase relationship between hippocampal place units and the EEG theta rhythm*, Hippocampus **3** (1993), 317–330.

[24] W. E. Skaggs, B. L. McNaughton, M. A. Wilson, and C. A. Barnes, *Theta phase precession in hippocampal neuronal populations and the compression of temporal sequences*, Hippocampus **6** (1996), 149172.

[25] A. A. Ghazanfar and M. A. Nicolelis, *Spatiotemporal properties of layer v neurons of the rat primary somatosensory cortex*, Cerebral Cortex **9** (1999), 348–361.

[26] L. Dolle, D. Sheynikhovich, B. Girard, B. Ujfalussy, R. Chavarriaga, and A. Guillot, *Analyzing interactions between cue-guided and place-based navigation with a computational model of action selection: influence of sensory cues and training*, in Proceedings of the 11th international conference on Simulation of adaptive behavior: from animals to animats, SAB'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 335–346.

[27] L.-E. Martinet, J.-B. Passot, B. Fouque, J.-A. Meyer, and A. Arleo, *Map-based spatial navigation: A cortical column model for action planning*, in Spatial Cognition VI. Learning, Reasoning, and Talking about Space, (C. Freksa, N. Newcombe, P. Grdenfors, and S. Wlfl, eds.), Vol. 5248 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 2008, pp. 39–55.

[28] L. Dolle, M. Khamassi, B. Girard, A. Guillot, and R. Chavarriaga, *Analyzing interactions between navigation strategies using a computational model of action selection*, LNAI **5248** (2008), 7186.

[29] R. Chavarriaga, T. Strosslin, D. Sheynikhovich, and W. Gerstner, *A computational model of parallel navigation systems in rodents*, Neuroinformatics **3** (2005), 223242.

[30] E. Uchibe and K. Doya, *Reinforcement learning with multiple heterogeneous modules: A framework for developmental robot learning*, International Conference on Development and Learning **4** (2005), 8792.

[31] B. Girard, D. Filliat, J. Meyer, A. Berthoz, and A. Guillot, *Integration of navigation and action selection functionalities in a computational model of cortico-basalthalamo-cortical loops*, Adapt. Behav. **13** (2005), 115130.

[32] A. Guazzelli, F. Corbacho, M. Bota, and M. Arbib, *Affordances, motivation, and the world graph theory*, Adapt. Behav. **6** (1998), 435471.

[33] C. Weber and J. Triesch, *Goal-directed feature learning*, Neural Networks, IEEE - INNS - ENNS International Joint Conference on **0** (2009), 3319–3326.

[34] E. Tolman and C. Honzik, *"Insight" in rats*, Univ. Calif. Publ. Psychol. **4** (1930), 215 – 232.

[35] J. Ferbinteanu and M. Shapiro, *Prospective and retrospective memory coding in the hippocampus*, Neuron **40** (2003), 1227–1239.

[36] A. P. Yonelinas, *The Nature of Recollection and Familiarity: A review of 30 Years of Research*, Journal of Memory and Language **46** (2002), 441–517.

[37] J. P. Aggletona, S. D. Vanna, C. Denbyb, S. Dixc, A. R. Mayesb, N. Robertsd, and A. P. Yonelinas, *Sparing of the familiarity component of recognition memory in a patient with hippocampal pathology*, Neuropsychologia **43** (2005), 1810–1823.

[38] A. D. Ekstrom, M. J. Kahana, J. B. Caplan, T. A. Fields, E. A. Isham, E. L. Newman, and I. Fried, *Cellular networks underlying human spatial navigation*, Nature **425** (2003), 184–187.

[39] H. Eichenbaum, *Hippocampus: Cognitive processes and neural representations that underlie declarative memory*, Neuron **44** (2004), 109–120.

[40] D. G. Mumby, *Perspectives on object recognition memory following hippocampal damage: lessons from studies on rats*, Behavioural Brain Research **127** (2001), 159–181.

[41] D. G. Mumby, S. Gaskin, M. J. Glenn, T. E. Scharamek, and H. Lehmann, *Hippocmapal damage and exploratory preferences in rats: memory for objects, place, and contexts*, Learning and Memory **9** (2002), 49–57.

[42] G. Norman and M. J. Eacott, *Dissociable effects of lesions to the perirhinal cortex and the postrhinal cortex on memory for context and objects in rats*, Behavioral Neuroscience **119** (2005), 557–566.

[43] N. S. Clayton, T. J. Bussey, and A. Dickinson, *Can animals recall the past and plan for the future*, Nature Reviews Neuroscience **4** (2003), 685–691.

[44] M. Day, R. Langston, and R. G. M. Morris, *Glutamate-receptor-mediated encoding and retrieval of paired-associate learning*, Nature **424** (2003), 205–209.

[45] C. Ergorul and H. Eichenbaum, *The hippocampus and memory for What, Where, and When*, Learning and Memory **11** (2004), 397–405.

[46] A. Baddeley, *The episodic buffer: A new component of working memory*, Trends in Cognitive Science **4** (2000), 417–423.

[47] E. Tulving, *Elements of episodic memory*, Oxford: Clarendon Press, 1983.

[48] Aristotle, *On Memory and Reminiscence*, 350BC.

[49] R. O. Hopkins and R. P. Kesner, *Item and order recognition memory in subjects with hypoxic brain injury*, Brain and Cognition **27** (1995), 180–201.

[50] H. J. Spiers, N. Burgess, T. Hartley, F. Vargha-Khadem, and J. OKeefe, *Bilateral hippocampal pathology impairs topographical and episodic memory but not visual pattern matching*, Hippocampus **11** (2001), 715–725.

[51] F. Vargha-Khadem, D. G. Gadin, K. E. Watkins, A. Connelly, W. V. Paesschen, and M. Mishkin, *Differential Effects of Early Hippocampal Pathology on Episodic and Semantic Memory*, Science **277** (1997), 376–380.

[52] J. J. Downes, A. R. Mayes, C. MacDonald, and N. M. Humkin, *Temporal order memory in patients with Korsakoffs syndrome and medial temporal amnesia*, Neuropsychologia **40** (2002), 853–861.

[53] L. McDonough, J. M. Mandler, R. D. McKee, and L. R. Squire, *The deferred imitation task as a nonverbal measure of declarative memory*, PNAS **92** (1995), 7580–7584.

[54] A. Adlam, F. Vargha-Khadem, M. Mishkin, and M. de Haan, *Deferred imitation of action dequences in developmental amnesia*, Journal of Cognitive Neuroscience **17** (2005), 240–248.

[55] R. C. Honey, A. Eatt, and M. Good, *Hippocampal lesions disrupt an associative mismatch process*, Journal of Neuroscience **18** (1998), 2226–2230.

[56] N. J. Fortin, K. L. Agster, and H. Eichenbaum, *Critical Role of the Hippocampus in Memory for Sequences of Events*, Nature Neuroscience **5** (2002), 458–462.

[57] R. P. Kesner, P. E. Gilbert, and L. A. Barua, *The role of the hippocampus in memory for the temporal order of a sequence of odors*, Behavioral Neuroscience **116** (2002), 286–290.

[58] R. E. Hampson, C. J. Heyser, and S. A. Deadwyler, *Hippocampal cell firing correlates of delayed-match-to-sample performance in the rat*, Behavioral Neuroscience **107** (1993), 715–739.

[59] H. Eichenbaum, P. Dudchencko, E. Wood, M. Shapiro, and H. Tanila, *The hippocampus, memory, and place cells: Is it spatial memory or a memory space*, Neuron **23** (1999), 209–226.

[60] W. Schultz, *Dopamine signals for reward value and risk: basic and recent data*, Behavioral and Brain Functions **6** (2010), 1–9.

[61] M. Bernard, J. D. Morrow, S. Taylor, S. Verzi, C. Vineyard, T. Caudell, N. Cohen, H. Eichenbaum, M. McDaniel, and P. Watson *Modeling Aspects of Human Memory for Scientific Study* Technical report, Sandia National Laboratories, 2009.

[62] M. J. Healy and T. P. Caudell, *Ontologies and Worlds in Category Theory: Implications for Neural Systems*, Axiomathes **16** (2006), 165–214.

[63] G. A. Carpenter and S. Grossberg, *A massively parallel architecture for a self-organizing neural pattern recognition machine*, Computer Vision, Graphics, and Image Processing **37** (1987), 54–115.

[64] G. A. Carpenter, S. Grossberg, and D. B. Rosen, *Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system*, Neural Networks **4** (1991), 759–771.

[65] M. Georgiopoulos, G. L. Heileman, and J. Huang, *Properties of learning related to pattern diversity in ART1*, Neural Networks **4** (1991), 751757.

[66] M. J. Healy and T. P. Caudell, *Discrete stack interval representations and fuzzy ART1*, Proceedings of the World Congress on Neural Networks **2** (1993), 82–91.

[67] ———, *A categorical semantic analysis of ART architectures*, IJCNN01: International Joint Conference of Neural Networks, Washington, DC **1** (2001), 38–43.

[68] D. Hebb, *The Organization of Behavior*, John Wiley, New York, NY, 1949.

[69] M. J. Healy, T. P. Caudell, and S. D. G. Smith, *A neural architecture for pattern sequence verification through inferencing*, Neural Networks, IEEE Transactions on **4** (1993), 9–20.

[70] T. P. Caudell and M. J. Healy, *Lateral Priming Adaptive Resonance Theory (LAPART)-2: Learning and Generalization*, CRC Press, Inc., Boca Raton, FL, 2000.

[71] M. J. Healy, *On the Semantics of Neural Networks*, Adaptive Neural Systems: The 1992 IR&D Technical Report, in T. P. Caudell, Ed. Technical Report BCS-CS-ACS-93-008 **1**.

[72] ———, *Continuous Functions and Neural Network Semantics*, Proceedings: The Second World Congress of Nonlinear Analysts **1**.

[73] S. E. Taylor, M. L. Bernard, T. P. Caudell, N. J. Cohen, M. J. Healy, J. D. Morrow, S. J. Verzi, C. M. Vineyard, and P. Watson, *Temporal semantics: an adaptive resonance theory approach*, in Proceedings of the International Joint Conference on Neural Networks, Institute of Electrical and Electronic Engineers, Atlanta, GA, USA, June 14–19 2009, pp. 2410–2416.

[74] M. J. Healy and T. P. Caudell *Temporal Sequencing via Supertemplates* Dspace-UNM EECE-TR-10-0001, University of New Mexico, 2006.

[75] L. Vila, *A Survey on Temporal Reasoning in Artificial Intelligence*, AI Communications **7** (1994), 4–28.

[76] S. Walczak, *Artificial neural network medical decision support tool: predicting transfusion requirements of ER patients*, Information Technology in Biomedicine, IEEE Transactions on **9** (2005), 468–474.

[77] S. Lawrence, C. L. Giles, and I. Fong, *Can recurrent neural networks learn natural language grammars*, in In Proceedings of the IEEE International Conference on Neural Networks IEEE Press, 1996, pp. 1853–1858.

[78] G. Kuhn, R. L. Watrous, and B. Ladendorf, *Connected recognition with a recurrent network*, Speech Communication **9** (1990), 41–48.

[79] E. A. Wan, *Time series prediction by using a connectionist network with internal delay lines*, in Time Series Prediction Addison-Wesley, 1994, pp. 195–217.

[80] L. Shastri and T. Fontaine, *Recognizing Hand-printed digit strings using modular spatio-temporal connectionist networks*, Connection Science **7** (1995), 211–235.

[81] R. Sun and C. Giles, *Sequence Learning: From Recognition and Prediction to Sequential Decision Making*, IEEE Intelligent Systems **16** (2001), 67–70.

[82] T. J. Sejnowksi and C. R. Rosenberg, *Parallel networks that learn to pronounce english text*, Complex Systems **1** (1987), 145–168.

[83] J. L. Elman, *Finding structure in time*, Cognitive Science **14** (1990), 179–211.

[84] S. Haykin, *Neural Networks a Comprehensive Foundation*, Prentice Hall, 1999.

[85] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, *Phoneme recognition using time-delay neural networks*, Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on **37** (1989), 328–339.

[86] C. Svarer, L. K. Hansen, and J. Larsen, *On design and evaluation of tapped-delay neural network architectures*, Neural Networks **1** (1993), 46–51.

[87] I. Sandberg and L. Xu, *Uniform approximation of multidimensional myopic maps*, IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications **44** (1997), 477–485.

[88] A. C. Tsoi and A. Back, *Discrete time recurrent neural network architectures: a unifying review*, Neurocomputing **15** (1997), 183–223.

[89] B. A. Pearlmutter, *Gradient calculations for dynamic recurrent neural networks: A survey*, IEEE Transactions on Neural Networks **6** (1995), 1212–1228.

[90] M. I. Jordan, *Attractor dynamics and parallelism in a connectionist sequential machine*, IEEE Press, Piscataway, NJ, USA, 1990.

[91] P. J. Werbos, *Backpropagation through time: what it does and how to do it*, Proceeding of the IEEE **78** (1990), 1550–1560.

[92] G. Bradski, G. A. Carpenter, and S. Grossberg, *Store working memory networks for storage and recall of arbitrary temporal sequences*, Biological Cybernetics **71** (1994), 469–480.

[93] ———, *Working memory networks for learning temporal order with application to three-dimensional visual object recognition*, Neural Computation **4** (1992), 270–286.

[94] R. Williams and D. Zipser, *A learning algorithm for continually running fully recurrent neural networks*, Neural Computation **1** (1989), 270–280.

[95] E. R. Kandel, J. H. Schwartz, and T. M. Jessell, *Principles of Neural Science*, McGraw-Hill, 2000.

[96] T. P. Caudell, Y. Xiao, and M. J. Healy, *eLoom and Flatland: specification, simulation and visualization engines for the study of arbitrary hierarchical neural architectures*, Neural Networks **16** (2003), 617–624.

[97] R. G. M. Morris, *Morris water maze*, Scholarpedia **3** (2008), 6315.

[98] ———, *Developments of a water-maze procedure for studying spatial learning in the rat*, Journal of Neuroscience Methods **11** (1984), 47–60.

[99] G. Han, F. M. Ham, and L. V. Fausett, *Fuzzy LAPART supervised learning through inferencing for stable category recognition*, Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the Third IEEE Conference on **1** (1994), 46–51.

[100] T. P. Caudell, C. T. Burch, M. Zengin, N. Gauntt, and M. J. Healy, *Retrospective learning of spatial invarients during object classification by embodied autonomous neural agents*, in Proceedings of the International Joint Conference on Neural Networks, Institute of Electrical and Electronic Engineers, San Jose, CA, USA, July 31 2011 - Aug. 5 2011 2011, pp. 2135 – 2142.

[101] R. W. Sperry, *Neurology and the Mind-Body Problem*, American Scientist **40** (1952), 291–312.

[102] W. Prinz, *Modes of linkage between perception and action*, in Cognition and Motor Processes, (W. Prinz and A. F. Sanders, eds.), Springer, Berlin, 1984, pp. 185–193.

[103] C. Cavina-Pratesi, R. W. Kentridge, C. A. Heywood, and A. D. Milner, *Separate processing of texture and form in the ventral stream: Evidence from fmri and visual agnosia*, Cerebral Cortex **20** (2010), 433–446.

[104] J. Vinberg and K. Grill-Spector, *Representation of shapes, edges, and surfaces across multiple cues in the human visual cortex*, Journal of Neurophysiology **99** (2008), 1380–1393.

[105] F. Fang, H. Boyaci, and D. Kersten, *Border ownership selectivity in human early visual cortex and its modulation by attention*, The Journal of Neuroscience **29** (2009), 460–465.

[106] S. Stigler, *Fisher and the 5% level*, CHANCE **21** (2008), 12–12 (10.1007/s00144-008-0033-3).

[107] D. A. Hamilton, G. K. Akers, M. P. Weisend, and R. J. Sutherland, *How Do Room and Apparatus Cues Control Navigation in the Morris Water Task? Evidence for Distinct Contributions to a Movement Vector*, Journal of Experimental Psychology: Animal Behaviour Processes **33** (2007), 100–114.

[108] R. G. M. Morris, *Spatial localisation does not require the presence of local cues*, Learning and Motivation **12** (1981), 239–260.

[109] S. M. Lane, *Categories for the Working Mathematician*, Springer, 1971.

[110] F. Lawvere and S. Schanuel, *Conceptual Mathematics: A First Introduction to Categories*, Cambridge University Press, 1991.

[111] C. J. Price and K. J. Friston, *Functional ontologies for cognition: The systematic definition of structure and function*, Cognitive Neuropsychology **22** (2005), 262–275.

[112] T. N. W. D. H. Hubel, *Brain and Visual Perception: The Story of a 25-year Collaboration*, Oxford University Press, 2005.

[113] J. M. Alonso, *Building Better Models of Visual Cortical Receptive Fields*, Neuron **46** (2005), 842–844.

[114] J. Bell, *Category Theory and the Foundations of Mathematics*, The British Journal for the Philosophy of Science **32** (1981), 349–358.

[115] M. J. Healy, R. D. Olinger, R. J. Young, S. E. Taylor, T. Caudell, and K. W. Larson, *Applying category theory to improve the performance of a neural architecture*, Neurocomputing **72** (2009), 3158 – 3173 (Hybrid Learning Machines (HAIS 2007) / Recent Developments in Natural Computation (ICNC 2007)).

[116] L. Davachi and A. D. Wagner, *Hippocampal contributions to episodic encoding, Insights from relational and item-based learning*, Journal of Neurophysiology **88** (2002), 982–990.

[117] K. S. Giovanello, M. Verfaellie, and M. M. Keane, *Disproportionate deficit in associative recognition relative to item recognition in global amnesia*, Cognitive, Affective, and Behavioral Neuroscience **3** (2003), 186–194.

[118] S. A. Small, A. S. Nava, G. M. Perera, R. DeLaPaz, R. Mayeux, and Y. Stern, *Circuit mechanisms underlying memory encoding and retrieval in the long axis of the hippocampal formation*, Nature Neuroscience **4** (2001), 442–449.

[119] M. M. Zeineh, S. A. Engel, P. M. Thompson, and S. Y. Brookheimer, *Dynamics of the hippocampus during encoding and retrieval of face-name pairs*, Science **299** (2003), 577–580.

[120] R. Sperling, E. Chua, A. Cocchiarella, E. Rand-Giovannetti, R. Poldrack, D. Schacter, and M. Albert, *Putting names to faces: Successful encoding of associative memories activates the anterior hippocampal formation*, NeuroImage **20** (2003), 1400–1410.

[121] K. S. Giovanello, D. M. Schnyer, and M. Verfaellie, *A critical role for the anterior hippocampus in relational memory: Evidence from an fMRI study comparing associative and item recognition*, Hippocampus **14** (2003), 5–8.

[122] P. Turriziani, L. Fadda, C. Caltagirone, and G. A. Carlesimo, *Recognition memory for single items and associations in amnesia patients*, Neuropsychologia **42** (2004), 426–433.

[123] L. R. Squire, C. E. L. Stark, and R. E. Clark, *The Medial Temporal Lobe*, Annual Review of Neuroscience **27** (2004), 279–306.

[124] N. J. Cohen, J. Ryan, C. Hunt, L. R. nad T. Wszalek, and C. Nash, *Hippocampal system and declarative (relational) memory: Summarizing the data from functional neuroimaging studies*, Hippocampus **9** (1999), 83–98.

[125] L. L. Eldridge, B. J. Knowlton, C. S. Furmanski, S. Y. Bookheimer, and S. A. Engel, *Remembering episodes: a selective role for the hippocampus during retrieval*, Nature Neuroscience **3** (2000), 1149 – 1152.

[126] E. A. Maguire, *Neuroimaging studies of autobiographical event memory*, Phil. Trans. R. Soc. Lond. B **356** (2001), 1441–1451.

[127] D. R. Addis, M. Moscovitch, A. P. Crawley, and M. P. McAndrews, *Recollective qualities modulate hippocampal activation during autobiographical memory retrieval*, Hippocampus **14** (2004), 752–762.

[128] K. Henke, B. Weber, S. Kneifel, H. G. Wieser, and A. Buck, *Human hippocampus associates information in memory*, PNAS **96** (1999), 5884–5889.

[129] L. Davachi, J. P. Mitchell, and A. D. Wagner, *Multiple routes to memory, Distinct medial temporal lobe processes build item and source memories*, PNAS **100** (2003), 2157–2162.

[130] C. Ranganath, A. P. Yonelinas, M. X. Cohen, C. J. Dy, S. M. Tom, and M. D. D'Esposito, *Dissociable correlates of recollection and familiarity with the medial temporal lobes*, Neuropsychologia **42** (2003), 2–13.

[131] H. Eichenbaum, A. R. Yonelinas, and C. Ranganath, *The medial temporal lobe and recognition memory*, Annual Review of Neuroscience **30** (2007), 123–152.

[132] E. Wood, P. Dudchencko, J. R. Robitsek, and H. Eichenbaum, *Hippocampal neurons encode information about different types of memory episodes occurring in the same location*, Neuron **27** (2000), 623–633.

[133] L. M. Frank, E. N. Brown, and M. Wilson, *Trajectory encoding in the hippocampus and entorhinal cortex*, Neuron **27** (2000), 169–178.

[134] W. James, *The Principles of Psychology*, New York: Holt, 1890.

[135] E. Tulving, *Episodic memory: From mind to brain*, Annual Review of Psychology **53** (2002), 1–25.

[136] M. Bunsey and H. Eichenbaum, *Conservation of hippocampal memory function in rats and humans*, Nature **379** (1996), 255–257.

[137] J. A. Dusek and H. Eichenbaum, *The hippocampus and memory for orderly stimulus relations*, PNAS **94** (1997), 7109–7114.

[138] P. Anderson, R. Morris, D. Amaral, T. Bliss, and J. O'Keefe, *The Hippocampus Book*, Oxford University Press, Oxford, 2007.

[139] J. K. Leutgeb, S. Leutgeb, M. B. Moser, and E. I. Moser, *Pattern separation in the dentate gyrus and CA3 of the hippocampus*, Science **315** (2007), 961–966.

[140] E. T. Rolls and R. P. Kesner, *A computational theory of hippocampal function, and empirical tests of the theory*, Progress in Neurobiology **79** (2006), 1–48.

[141] K. Nakazawa, L. D. Sun, M. C. Quirk, L. Rondi-Reig, M. A. Wilson, and S. Tonegawa, *Hippocampal CA3 NMDA Receptors Are Crucial for Memory Acquisition of One-Time Experience*, Neuron **38** (2003), 305–315.

[142] R. C. O'Reilly and J. L. McClelland, *Hippocampal conjunctive encoding, storage, and recall: Avoiding a trade-off*, Hippocampus **4** (1994), 661682.

[143] T. Kohonen, *The self-organizing map. Proc Inst Electrical Electronics*, Proc IEEE **78** (1990), 14641480.

[144] M. J. Healy, *A topological semantics for rule extraction with neural networks*, Connection Science **11** (1999), 91–113.

[145] A. Bakker, C. B. Kirwan, M. Miller, and C. E. L. Stark, *Pattern Separation in the Human Hippocampal CA3 and Dentate Gyrus*, Science **319** (2008), 1640–1642.

[146] M. E. Hasselmo, E. Schnell, and E. Barkai, *Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3*, Journal of Neuroscience **15** (1995), 5249–5262.

[147] D. E. Hannula, J. D. Ryan, D. Tranel, and N. J. Cohen, *Rapid onset relational memory effects are evident in eye movement behavior, but not in hippocampal amnesia*, Journal of Cognitive Neuroscience **19** (2007), 1690–1705.

[148] A. Preston, Y. Shrager, N. M. Dudukovic, and J. D. E. Gabrieli, *Hippocampal contribution to the novel use of relational information in declarative memory*, Hippocampus **14** (2004), 148–152.

[149] T. M. Cover, *Elements of Information Theory*, J. Wiley, Hoboken, N.J., 2005.

[150] J. K. Leutgeb, S. Leutgeb, A. Treves, R. Meyer, C. A. Barnes, B. L. McNaughton, M. B. Moser, and E. I. Moser, *Progressive Transformation of Hippocampal Neuronal Representations in Morphed Environments*, Neuron **48** (2005), 345–358.