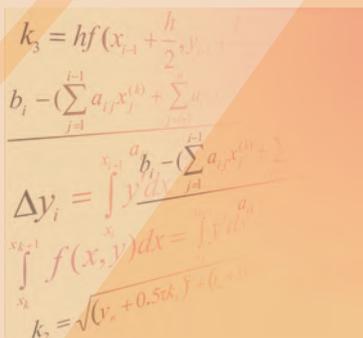


Proceedings of 2013 BMI the Second International Conference on Brain-Mind

July 27-28
East Lansing, Michigan USA



BMI Press

ISBN 978-0-9858757-0-1

US\$35

Contents

1	Messages From the Chairs	iii
2	Committees	iv
2.1	Advisory Committee.....	iv
2.2	Program Committee.....	iv
2.3	Conference Committee.....	iv
2.4	MSU Steering Committee.....	iv
3	Invited Talks	I
3.1	Neural Coding and Decoding: An Overview of the Neuroscience and Neurophysiology behind Intracortical Brain-Computer Interfaces <i>Beata Jarosiewicz</i>	I
3.2	Resting-State fMRI and Applications <i>David C. Zhu</i>	II
3.3	Examining the Effects of Avatar-body Schema Integration <i>Rabindra (Robby) Ratan</i>	III
3.4	BrainGate: Toward the Development of Brain-Computer Interfaces for People with Paralysis <i>Beata Jarosiewicz</i>	IV
3.5	Obama’s Brain Initiative and Resistance from the Status Quo <i>Juyang Weng</i>	V
3.6	Representation of Attentional Priority in Human Cortex <i>Taosheng Liu</i>	VI
4	Paper	1
4.1	Full paper	1
4.1.1	Skull-closed Autonomous Development: WWN-7 Dealing with Scales <i>Xiaofeng Wu, Qian Guo, Yuekai Wang and Juyang Weng</i>	1
4.1.2	Serious Game Modeling of Caribou Behavior Across Lake Huron using Cultural Algorithms and Influence Maps <i>James Fogarty, J. O’Shea, Xiangdong Sean Che, Robert G. Reynolds</i>	9

4.1.3	Establish the Three Theorems: DP Optimally Self-Programs Logics Directly from Physics <i>Juyang Weng</i>	19
4.1.4	Neural Modulation for Reinforcement Learning in Developmental Networks Facing an Exponential No. of States <i>Hao Ye, Xuanjing Huang, and Juyang Weng</i>	27
4.1.5	The Ontology of Consciousness Eugene <i>M. Brooks, M.D.</i>	35
4.2	Abstract	40
4.2.1	Motor neuron splitting for efficient learning in Where-What Network <i>Zeja Zheng, Kui Qian, Juyang Weng, and Zhengyou Zhang</i>	40
4.2.2	Neuroscientific Critique of Depression as Adaptation <i>Gonzalo Munevar, Donna Irvan</i>	41
5	Sponsors	42

Messages From the Chairs

2013 is the second year of the Brain-Mind Institute (BMI) and the International Conference on Brain-Mind (ICBM).

April 2, 2013, President Barack Obama announced his Brain Initiative. The European Union has announced the Human Brain Project. China is preparing its own brain project. Understanding how the brain works is one of the last frontiers of the human race. The era where humans can understand how their brains work seems to have arrived, although any understanding of the nature is always an approximation. When a model can predict observed data well, the model is a good approximation in terms of the observed data.

The subject of brain-mind is closely related to all activities of the human race. For this reason, BMI started an earlier platform that treats every human activity as a part of science, including, but not limited to, biology, neuroscience, psychology, computer science, electrical engineering, mathematics, intelligence, life, laws, policies, societies, and politics. The scientific community faces great opportunities and challenges, ranging from communication to education, to research and to outreach. BMI tries to serve the scientific community and public.

After offering BMI 821 Biology for Brain-Mind Research, BMI 821 Neuroscience for Brain-Mind Research, and BMI 871 Computational Brain-Mind in 2012, this year BMI offered BMI 871 Computational Brain-Mind and BMI 831 Cognitive Science for Brain-Mind Research. We would like to thank Fudan University for hosting the BMI 871 classes 2012 and 1013 and Michigan State University for hosting the BMI 811 and BMI 821 in 2012, and BMI 831 in 2013. BMI courses were offered in two forms, live classes and distance-learning classes. BMI plans to host BMI courses and ICBM at more international locations in the future.

As a multi-disciplinary communication platform for exchanging latest research ideas and results, ICBM is an integrated part of the BMI program. ICBM 2013 includes invited talks, talks from submitted papers, and talks from submitted abstracts. From this year, ICBM talks will be video recorded and available publicly through the Internet.

The brain-mind subjects are highly multidisciplinary. The BMI Program Committee tries to be open-minded in review of submissions. This open-mindedness is necessary for the broad nature of brain-mind education and research.

Welcome to East Lansing!

Jiaguo Qi, Program Co-Chair

George Stockman, Program Co-Chair

Yang Wang, Program Co-Chair

Juyang Weng, General Chair

Committees

Advisory Committee

- Dr. Stephen Grossberg, Boston University, USA
- Dr. James McClelland, Stanford University, USA
- Dr. James Olds, George Mason University, USA
- Dr. Linda Smith, Indiana University, USA
- Dr. Mriganka Sur, Massachusetts Institute of Technology (MIT), USA

Program Committee

- Dr. Yiannis Aloimonos, University of Maryland, USA
- Dr. Minoru Asada, Osaka University, Japan
- Dr. Luis M. A. Betterncourt, Los Alamos National Laboratory, USA
- Dr. Angelo Cangelosi, University of Plymouth, UK
- Dr. Shantanu Chakrabartty, Michigan State University, USA
- Dr. Yoonsuck Choe, Texas A&M University, College Station, USA
- Dr. Wlodzislaw Duch, Nicolaus Copernicus University, Poland
- Dr. Zhengping Ji, Los Alamos National Laboratory, USA
- Dr. Yaochu Jin, University of Surrey, UK
- Dr. Pinaki Mazumder, University of Michigan, USA
- Dr. Yan Meng, Stevens Institute of Technology, Hoboken, USA
- Dr. Ali A. Minai, University of Cincinnati, USA
- Dr. Jun Miao, Chinese Academy of Sciences, China
- Dr. Thomas R. Shultz, McGill University, Canada
- Dr. Linda Smith, Indiana University, USA
- Dr. Juyang Weng, Michigan State University, USA
- Dr. Xiaofeng Wu, Fudan University, China
- Dr. Ming Xie, Nanyang Technological University, Singapore
- Dr. Xiangyang Xue, Fudan University, China
- Dr. Chen Yu, Indiana University, USA
- Dr. Cha Zhang, Microsoft Research, USA

- Dr. Shuqing Zeng, GM Research and Development, USA

Conference Committee

- Dr. James H. Dulebohn, Michigan State University, USA
- Dr. Jianda Han, Shengyang Institute of Automation, China
- Dr. Kazuhiko Kawamura, Vanderbilt University, USA
- Dr. Minh Lee, Kyungpook National University, Korea
- Dr. Gonzalo Munevar, Lawrence Technological University, USA
- Dr. Danil Prokhorov, Toyota Research Institute of America, USA
- Dr. Robert Reynolds, Wayne State University, USA
- Dr. Katharina J. Rohlfing, Bielefeld University, Germany
- Dr. Matthew Schlesinger, Southern Illinois University, USA
- Dr. Jochen Triesch, Frankfurt Institute for Advanced Studies, Germany
- Dr. Hiroaki Wagatsuma, Kyushu Institute of Technology, Japan

MSU Steering Committee

- Dr. Alan Beretta, Professor, Michigan State University
- Dr. Andrea Bozoki, Michigan State University
- Dr. Jay P. Choi, Michigan State University
- Dr. Lynwood G. Clemens, Michigan State University
- Dr. Kathy Steece-Collier, Michigan State University
- Dr. Steve W. J. Kozlowski, Michigan State University
- Dr. Jiaguo Qi, Michigan State University
- Dr. Frank S. Ravitch, Michigan State University
- Dr. Fathi Salem, Michigan State University
- Dr. George Stockman, Michigan State University
- Dr. Yang Wang, Michigan State University
- Dr. Juyang Weng, Michigan State University
- Dr. David C. Zhu, Michigan State University

Invited Talks

Neural Coding and Decoding: An Overview of the Neuroscience and Neurophysiology behind Intracortical Brain-Computer Interfaces.

Beata Jarosiewicz, Brown University

Abstract

Conditions such as brainstem stroke, spinal cord injury, and amyotrophic lateral sclerosis (ALS) can disconnect the brain from the rest of the body, leaving the person awake and alert but unable to move. Conventional assistive devices for people with severe motor disabilities are inherently limited, often relying on residual motor function for their use. Brain-computer interfaces (BCIs) aim to provide a more powerful signal source by tapping into the rich information content that is still available in the person's brain activity. A crucial component of BCIs is the ability to record neural activity and decode information from it. In this lecture, I will give an overview of the neuroscience and neurophysiology behind neural coding and decoding, drawing examples from well-studied brain systems such as the visual system, the hippocampal place cell system, and the motor system.

Short Biography

Dr. Jarosiewicz is an Investigator in Neuroscience at Brown University in Providence, RI. She received her Ph.D. in 2003 in the laboratory of William Skaggs at the University of Pittsburgh and the Center for the Neural Basis of Cognition, characterizing the activity of place cells in a novel physiological state in the rat hippocampus. She did postdoctoral research with Dr. Andrew Schwartz at the University of Pittsburgh, where she studied neural plasticity in non-human primates using brain-computer interfaces, and then with Dr. Mriganka Sur at MIT, where she used 2-photon calcium imaging to characterize the properties of ferret visual cortical neurons with known projection targets. She joined the BrainGate research team at Brown University in 2010, where she is applying her neuroscience expertise to help develop practical intracortical brain-computer interfaces for people with severe motor disabilities.

Resting-State fMRI and Applications

David C. Zhu, Michigan State University

Abstract

Recently, resting state-fMRI (rs-fMRI) has emerged as an effective way to investigate brain networks. In this technique, fMRI data is acquired when an individual is asked to do nothing but stay awake while lying in the MRI scanner. The rs-fMRI technique emerged from the phenomena that approximately 95% of the brain's metabolism occurs because of spontaneous neuronal activity. The blood-oxygen-level-dependent (BOLD) fMRI signal indirectly measures the spontaneous neural activity. Therefore, the correlation of BOLD signal time courses between two brain regions at rest infers the functional connectivity between them. The fMRI signals from random brain activity are removed from correlations over a reasonably lengthy fMRI time course. Recent studies have demonstrated the potential applications of rs-fMRI in understanding the functional connectivity in the brains of both healthy individuals and neurological patients. In this talk, I will describe the underlying mechanism of resting-state fMRI and discuss potential applications.

Short Biography

I have 17 years of MRI research and development experience, including 13 years after I completed my Ph.D. degree in biomedical engineering at University of California, Davis. I developed my expertise in MRI physics and engineering during my graduate research and my subsequent work in GE Healthcare. After spending three years at University of Chicago as a research faculty member, I joined the faculty at Michigan State University in 2005. With other faculty members, we developed the Cognitive Imaging Research Center, and I have been supporting its growth in a role of an MRI physicist and the lead of the support team. I currently serve as an MRI physicist for the Cognitive Imaging Research Center (CIRC), and the Departments of Radiology and Psychology at Michigan State University. I also serve on the faculty of MSU Neuroscience and Cognitive Science programs. I am responsible for the technical aspect of CIRC. I have collaborated extensively with MSU psychologists and neuroscientists who are interested in using MR neuroimaging methods. Two of my research focuses are to study the functional and structural connectivity of brains affected by Alzheimer's disease and by concussion.

Examining the Effects of Avatar-body Schema Integration

Rabindra (Robby) Ratan, Michigan State University

Abstract

There is a growing body of research about the outcomes of using virtual avatars (and other mediated self-representations). For example, the Proteus Effect suggests that people behave in ways that conform to their avatars' characteristics, even after avatar use, e.g., using taller avatars leads to more social confidence (Yee & Bailenson, 2007). But there is little research on how the cognitive experience of using the avatar influences such effects. This talk will argue that just as humans are able to integrate complex tools into body schema (Gallivan et al., 2013), we can also integrate avatars into body schema. Doing so requires a high level of proficiency controlling the avatar, which many people attain through modern gaming interfaces. I argue that such integration of the avatar into body schema fundamentally modifies the effects of using the avatar. Somewhat counter-intuitively, my research suggests that avatar-body schema integration weakens post-use Proteus effects because it detracts from relevance of the avatar's identity characteristics and also augments the salience of disconnection from the avatar after use. I will present supporting data from an experiment using psychophysiological measurements, describe a second similar experiment that is currently underway, and discuss possible experimental designs with functional MRI to address this research question.** I should note that I am a media-technology scholar, not a neuroscientist nor an expert in the neural mechanisms of tool-body schema integration, so I welcome feedback from the neuroscience community and am open to collaboration with interested parties.

Short Biography

Rabindra ("Robby") Ratan's research focuses primarily on the psychological experience of media use, with an emphasis on video games and other interactive environments (e.g., the road) that include mediated self-representations (e.g., avatars, automobiles). He is particularly interested in how different facets of mediated self-representations (e.g., gender, social identity) influence the psychological experience of media use, and how different facets of this psychological experience (e.g., avatar-body schema integration, identification) affect a variety of outcomes, including cognitive performance, learning, health-related behaviors (e.g., food choice, driving aggression), and prejudicial/prosocial attitudes.

Methodologically, his work mostly includes experiments that utilize video game-based stimuli with psychophysiological and survey measures, as well as analyses of behavior-log databases (from games and other media) linked to surveys provided by users. Most recently, he has been developing games (with game-design students from the TISM department) that include potential experimental manipulations relating to research questions of interest (e.g., the effect of avatar characteristics on learning and post-play motivations). He plans to use these games in his studies as well as to release them to the general public.

BrainGate: Toward the Development of Brain-Computer Interfaces for People with Paralysis.

Beata Jarosiewicz, Brown University

Abstract

Our group, BrainGate, aims to restore independence to people with severe motor disabilities by developing brain-computer interfaces (BCIs) that decode movement intentions from spiking activity recorded from microelectrode arrays implanted in motor cortex of people with tetraplegia. This technology has already allowed people with tetraplegia to control a cursor on a computer screen, a robotic arm, and other prosthetic devices simply by imagining movements of their own arm. In this lecture, I will present an overview of BrainGate's ongoing research efforts, and I will discuss my efforts toward bringing the system closer to clinical utility by automating the self-calibration of the decoder during practical BCI use.

Short Biography

Dr. Jarosiewicz is an Investigator in Neuroscience at Brown University in Providence, RI. She received her Ph.D. in 2003 in the laboratory of William Skaggs at the University of Pittsburgh and the Center for the Neural Basis of Cognition, characterizing the activity of place cells in a novel physiological state in the rat hippocampus. She did postdoctoral research with Dr. Andrew Schwartz at the University of Pittsburgh, where she studied neural plasticity in non-human primates using brain-computer interfaces, and then with Dr. Mriganka Sur at MIT, where she used 2-photon calcium imaging to characterize the properties of ferret visual cortical neurons with known projection targets. She joined the BrainGate research team at Brown University in 2010, where she is applying her neuroscience expertise to help develop practical intracortical brain-computer interfaces for people with severe motor disabilities.

Obama's Brain Initiative and Resistance from the Status Quo

Juyang Weng, Michigan State University

Abstract

In this talk, I will first provide an overview about the challenges that Obama's Brain Initiative raised to the US government and the scientific community. It is well recognized that neuroscience has been productive but is rich in data and poor in theory. Still, it is natural but shortsighted for a government officer to approach only well-known experimental neuroscientists for advice on the Brain Initiative. I argue that it is impractical for experimental neuroscientists to come up with a comprehensive computational brain theory, because brain activities are numerical and highly analytical, which require extensive knowledge in analytical disciplines such as computer science, electrical engineering and mathematics. However, the status quo in those analytical disciplines still fall behind greatly, not only in terms of knowledge required to address the problems of the Brain Initiative, but also in terms of the persistent resistance toward brain subjects cause by the very human nature. Currently, almost all scholars, whether on the natural intelligence side or the artificial intelligence side, are highly skeptical about, and resisting, any comprehensive computational brain theory. The human race in its modern time is repeating the objections to new science like those toward Charles Darwin's theory of evolution. Open-minded communication and debates seem to be necessary to avoid taxpayer's money being unwisely spent on only incremental work.

Short Biography

Juyang (John) Weng is a professor at the Department of Computer Science and Engineering, the Cognitive Science Program, and the Neuroscience Program, Michigan State University, East Lansing, Michigan, USA. He received his BS degree from Fudan University in 1982, his MS and PhD degrees from University of Illinois at Urbana-Champaign, 1985 and 1989, respectively, all in Computer Science. His research interests include computational biology, computational neuroscience, computational developmental psychology, biologically inspired systems, computer vision, audition, touch, behaviors, and intelligent robots. He is the author or coauthor of over two hundred fifty research articles. He is a Fellow of IEEE, an editor-in-chief of International Journal of Humanoid Robotics and an associate editor of the new IEEE Transactions on Autonomous Mental Development. He has chaired and co-chaired some conferences, including the NSF/DARPA funded Workshop on Development and Learning 2000 (1st ICDL), 2nd ICDL (2002), 7th ICDL (2008), 8th ICDL (2009), and INNS NNN 2008. He was the Chairman of the Governing Board of the International Conferences on Development and Learning (ICDLs) (2005-2007, <http://cogsci.ucsd.edu/~triesch/icdl/>), chairman of the Autonomous Mental Development Technical Committee of the IEEE Computational Intelligence Society (2004-2005), an associate editor of IEEE Trans. on Pattern Recognition and Machine Intelligence, an associate editor of IEEE Trans. on Image Processing.

Representation of Attentional Priority in Human Cortex

Taosheng Liu, Michigan State University

Abstract

Humans can flexibly select certain aspects of the sensory information for prioritized processing. How such selection is achieved in the brain remains a major topic in cognitive neuroscience. In this talk, I will examine the neural mechanisms underlying both spatial and non-spatial selection. I will review evidence that space-based selection is controlled by dorsal frontoparietal areas that encode spatial priority in topographic maps, whereas feature- and object-based selection also rely on similar brain areas. These areas modulate neural activity in early visual areas to enhance the representation of task-relevant information. Furthermore, a recent study from our group found that spatial and feature-based priority forms a hierarchical structure in frontoparietal areas such that similar selection demands recruit similar neural activity patterns. These results suggest that the representation of attentional priority utilizes a computationally efficient organization to support flexible top-down control.

Short Biography

Taosheng Liu received his PhD in Cognitive Psychology from Columbia University and postdoctoral training at the Johns Hopkins University and New York University. He is now an Assistant Professor in the Department of Psychology at Michigan State University. Taosheng Liu's research interests are in the cognitive neuroscience of visual perception and attention, working memory, and decision making. His main experimental techniques include using psychophysics and eyetracking to measure behavior and using functional magnetic resonance imaging (fMRI) to measure human brain activity. Current research in his lab focuses on the representation of feature- and object-based attentional priority in the brain, how attention affects perception, and the neural mechanism of value-based decision making. More information can be found online at <http://psychology.msu.edu/LiuLab>.

Skull-closed Autonomous Development: WWN-7 Dealing with Scales

Xiaofeng Wu, Qian Guo, Yuekai Wang and Juyang Weng

Abstract—The Where-What Networks (WWNs) consist of a series of embodiments of a general-purpose brain-inspired network called Developmental Network (DN). WWNs model the dorsal and ventral two-way streams that converge to, and also receive information from, specific motor areas in the frontal cortex. Both visual detection and visual recognition tasks were trained concurrently by such a single, highly integrated network, through autonomous development. By “autonomous development”, we mean that not only that the internal (inside the “skull”) self-organization is fully autonomous, but the developmental program that regulates the growth and adaptation of computational network is also task non-specific. This paper focused on the “skull-closed” WWN-7 in dealing with different object scales. By “skull-closed”, we mean that the brain inside the skull, except the brain’s sensory ends and motor ends, is off limit throughout development to all teachers in the external physical environment. The concurrent presence of multiple learned concepts from many object patches is an interesting issue for such developmental networks in dealing with objects of multiple scales. Moreover, we will show how the motor initiated expectations through top-down connections as temporal context assist the perception in a continuously changing physical world, with which the network interacts. The inputs to the network are drawn from continuous video taken from natural settings where, in general, everything is moving while the network is autonomously learning.

I. INTRODUCTION

In the recent years, much effort has been spent on the field of artificial intelligence (AI) [1]. As the field of AI is inspired by human intelligence, more and more artificial intelligent models proposed are inspired by the brain to different degrees [2]. General objects recognition and attention is one of the important issues among the field of AI. And since human vision systems can accomplish such tasks quickly, mimicking the human vision systems is thought as one possible approach to address this open yet important vision problem.

In the primate vision system, two major streams have been identified [3]. The ventral stream involving V1, V2, V4 and the inferior temporal cortex is responsible for the cognition of shape and color of objects. The dorsal stream involving V1, V2, MT and the posterior parietal cortex takes charge of spatial and motion cognition. Put simply, the ventral

stream (what) is sensitive to visual appearance and is largely responsible of object recognition. The dorsal (where and how) is sensitive to spatial locations and processes motion information.

With the advances of the studies on visual cortex in physiology and neuroscience, several cortex-like network models have been proposed. One Model is HMAX, introduced by Riesenhuber and Poggio [4], [5]. This model is a hierarchical system that closely follows the organization of visual cortex and builds an increasingly complex and invariant feature representation by alternating between a template matching and a maximum pooling operation. In the simplest form of the model, it contains four layers, which are S_1 , C_1 , S_2 , C_2 from bottom to top. S_1 units corresponding to the classical simple cells in primary visual cortex (V1) [6] take the form of Gabor functions to detect the features with different orientations and scales, which have been shown to provide a good model of cortical simple cell receptive fields. C_1 units corresponding to cortical complex cells which show some tolerance to shift and size takes the maximum over a local spatial neighbourhood of the afferent S_1 units from the previous layer with the same orientation and scale band (each scale band contains two adjacent Gabor filter sizes). S_2 units measure the match between a stored prototype P_i and the input image at every position and scale using radial basis function (RBF). C_2 units takes a global maximum over each S_2 type (each prototype P_i), i.e., only keep the value of the best match and discard the rest. Thus C_2 responses are shift- and scale-invariant, which are then passed to a simple linear classifier (e.g., SVM). In summary, HMAX is a feed-forward network using unsupervised learning, which only models the ventral pathway in primate vision system while the location information is lost, to implement the feature extraction and combination. And a classifier (e.g., SVM) is a must for the task of object recognition, which means the feature extraction and classification are not integrated in a single network.

Different from HMAX, WWNs introduced by Juyang Weng and his co-workers is a biologically plausible developmental model [7], [8], [9] designed to integrate the object recognition and attention namely, what and where information in the ventral stream and dorsal stream respectively. It uses both feedforward (bottom-up) and feedback (top-down) connections. Moreover, multiple concepts (e.g., type, location, scale) can be learned concurrently in such a single network through autonomous development. That is to say, the feature representation and classification are highly integrated in a single network.

WWN has six versions. WWN-1 [10] can realize object

Qian Guo, Yuekai Wang and Xiaofeng WU are with State Key Lab. of ASIC & System, Fudan University, Shanghai, 200433, China and Department of Electronic Engineering, Fudan University, Shanghai, 200433, China, (email: {09300720079, 10210720110, xiaofengwu} @fudan.edu.cn); Juyang Weng is with School of Computer Science, Fudan University, Shanghai, 200433, China and Department of Computer Science and Engineering, Michigan State University, East Lansing, Michigan, 48824, USA, (email:weng@cse.msu.edu); This work was supported by Fund of State Key Lab. of ASIC & System (11M-S008) and the Fundamental Research Funds for the Central Universities to XW, Changjiang Scholar Fund of China to JW.

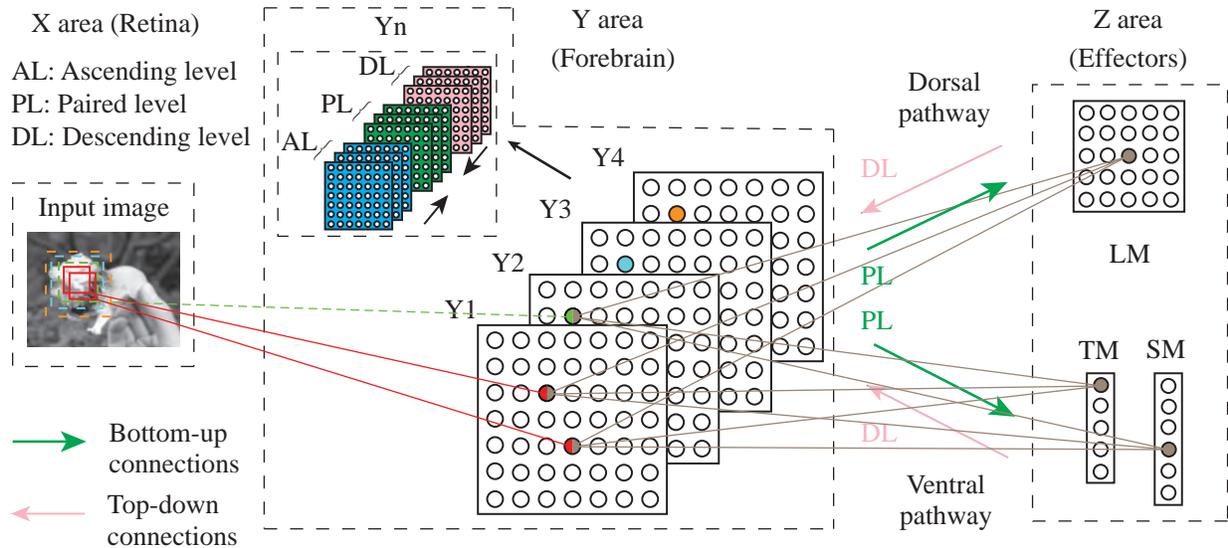


Fig. 1: The structure of WWN-7. The squares in the input image represent the receptive fields perceived by the neurons in the different Y areas. The red solid square corresponds to Y_1 , the green dashed square with the smallest interval corresponds to Y_2 , the blue and orange one with larger interval corresponds to Y_3 and Y_4 , respectively. Three linked neurons are firing, activated by the stimuli.

recognition in complex backgrounds performing in two different selective attention modes: the top-down position-based mode finds a particular object given the location information; the top-down object-based mode finds the location of the object given the type. But only 5 locations were tested. WWN-2 [11] can additionally perform in the mode of free-viewing, realizing the visual attention and object recognition without the type or location information and all the pixel locations were tested. WWN-3 [12] can deal with multiple objects in natural backgrounds using arbitrary foreground object contours, not the square contours in WWN-1. WWN-4 used and analyzed multiple internal areas [13]. WWN-5 is capable of detecting and recognizing the objects with different scale in the complex environments [14]. WWN-6 [15] has implemented truly autonomous skull-closed [16], which means that the “brain” inside the skull is not allowed to supervised directly by the external teacher during training and the internal connections are capable of self-organizing autonomously and dynamically (including on and off), meaning more closer to the mechanisms in the brain.

In this paper, a new version of WWN, named WWN-7, is proposed. Compared with the prior versions, especially recent WWN-5 and WWN-6 [17], WWN-7 have at least three innovations described below:

- WWN-7 is skull-closed like WWN-6, but it can deal with multiple object scales.
- WWN-7 is capable of dealing with multiple object scales like WWN-5, but it is truly skull-closed.
- WWN-7 has the capability of temporal processing, and uses the temporal context to guide visual tasks.

In the remainder of the paper, Section II overviews the architecture and operation of WWN-7. Section III presents some important concepts and algorithms in the network. Experimental results are reported in Section IV. Section V

gives the concluding remarks.

II. NETWORK OVERVIEW

In this section, the network structure and the overall scheme of the network learning are described.

A. Network Structure

The network (WWN-6) is shown as Fig. 1 which consists of three areas, X area (sensory ends/sensors), Y area (internal brain inside the skull) and Z area (motor ends/effectors). The neurons in each area are arranged in a grid on a 2D plane, with equal distance between any two adjacent (non-diagonal) neurons.

X acts as the retina, which perceives the inputs and sends signals to internal brain Y . The motor area Z serves as both input and output. When the environment supervises Z , Z is the input to the network. Otherwise, Z gives an output vector to drive effectors which act on the real world. Z is used as the hub for emergent concepts (e.g., goal, location, scale and type), abstraction (many forms mapped to one equivalent state), and reasoning (as goal-dependant emergent action). In our paradigm, three categories of concepts emerge in Z supervised by the external teacher, the location of the foreground object in the background, the type and the scale of this foreground object, corresponding to Location Motor (LM), Type Motor (TM) and Scale Motor (SM).

Internal brain Y is like a limited-resource “bridge” connecting with other areas X and Z as its two “banks” through 2-way connections (ascending and descending). Y is inside the closed skull, which is off limit to the teachers in the external environments. In WWN-7, there are multiple Y areas with different receptive fields, shown as $Y_1, Y_2, Y_3, Y_4, \dots$ in Fig. 1. Thus the neurons in different Y areas can represent the object features of multiple scales. Using a pre-screening area for each source in each Y area, before integration,

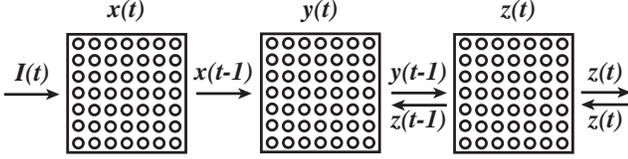


Fig. 2: Architecture diagram of a three-layer network. $\mathbf{I}(t)$ is an image from a discrete video sequence at time t . $\mathbf{x}(t)$, $\mathbf{y}(t)$ and $\mathbf{z}(t)$ is the response of the area X , Y and Z at time t , respectively. The update of each area is asynchronous, which means that at time t , $\mathbf{x}(t)$ is the response corresponding to $\mathbf{I}(t)$ (suppose no time delay, and in our experiment, $\mathbf{x}(t) = \mathbf{I}(t)$), $\mathbf{y}(t)$ is the response with the input $\mathbf{x}(t-1)$ and $\mathbf{z}(t-1)$, and similarly, $\mathbf{z}(t)$ is the response with the input $\mathbf{y}(t-1)$. Based on this analysis, $\mathbf{z}(t)$ is corresponding to the input image frame $\mathbf{I}(t-2)$, i.e., two-frame delay.

results in three laminar levels: the ascending level (AL) that pre-screens the bottom-up input, the descending level (DL) that pre-screens the top-down input and paired level (PL) that combines the outputs of AL and DL. In this model, there exist two pathways and two connections. Dorsal pathway refers to the stream $X \Rightarrow Y \Rightarrow LM$, while ventral pathway refers to $X \Rightarrow Y \Rightarrow TM$ and SM , where \Rightarrow indicates that each of the two directions has separate connections. That is to say, X provides bottom-up input to AL, Z gives top-down input to DL, and then PL combines these two inputs.

The dimension and representation of X and Z areas are hand designed based on the sensors and effectors of the robotic agent or biologically regulated by the genome. Y is skull-closed inside the brain, not directly accessible by the external world after the birth.

B. General Processing Flow of the Network

For explaining the general processing flow of the Network, Fig. 1 is simplified into a three-layer network shown as Fig. 2, representing X , Y and Z respectively.

Suppose that the network operates at discrete times $t = 1, 2, \dots$. This series of discrete time can represent any network update frequency. Denote the sensory input at time t to be \mathbf{I}_t , $t = 1, 2, \dots$, which can be considered as an image from a discrete video sequence. At time $t = 1, 2, \dots$, for each A in $\{X, Y, Z\}$ repeat:

- 1) Every area A computes its area function f , described below,

$$(\mathbf{r}', N') = f(\mathbf{b}, \mathbf{t}, N)$$

where \mathbf{r}' is the new response vector of A , \mathbf{b} and \mathbf{t} is the bottom-up and top-down input respectively.

- 2) For every area A in $\{X, Y, Z\}$, A replaces: $N \leftarrow N'$ and $\mathbf{r} \leftarrow \mathbf{r}'$. If this replacement operation is not applied, the network will not do learning anymore.

The update of each area described above is asynchronous [18] shown as the table, which means for each area A in $\{X, Y, Z\}$ at time t , the input is the response of the corresponding area at time $t-1$. For example, the bottom-up and top-down input to Y area at time t is the response of

Time t	0	1	2	3	4	5	6	7	8	9	10
$\mathbf{z}(t)$: su	B	B	α	*	α	β	*	β	α	*	*
$\mathbf{z}(t)$: em	-	-	?	α	?	?	β	?	?	α	α
$\mathbf{y}(t)$: z	-	B	B	α	α	α	β	β	β	α	α
$\mathbf{y}(t)$: x	-	α	α	α	β	β	β	α	α	α	β
$\mathbf{x}(t)$	α	α	α	β	β	β	α	α	α	β	β

TABLE I: Time sequence for an example: the teacher wants to teach a network to recognize two foreground objects α and β . “B” represents the concept of no interested foreground objects in the image(i.e., neither α nor β). “em”: emergent if not supervised; “su”: supervised by the teacher. “*” means free. “-” means not applicable.

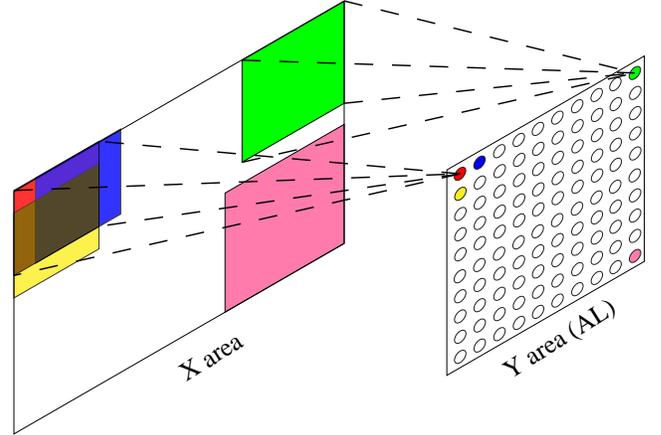


Fig. 3: The illustration of the receptive fields of neurons

X and Z area at time $t-1$ respectively. Based on such an analysis, the response of Z at time t is the result of the both $\mathbf{x}(t-2)$ and $\mathbf{z}(t-2)$. This mechanism of asynchronous update is different from the synchronous update in WVN-6, where the time of computation of each area was not considered.

In the remaining discussion, $\mathbf{x} \in X$ is always supervised. The $\mathbf{z} \in Z$ is supervised only when the teacher chooses. Otherwise, \mathbf{z} gives (predicts) effector output.

According to the above processing procedure (described in details in section III), an artificial Developmental Program (DP) is handcrafted by a human to short cut extremely expensive evolution. The DP is task-nonspecific as suggested for the brain in [19], [20] (e.g., not concept-specific or problem-specific).

III. CONCEPTS AND ALGORITHMS

A. Inputs and Outputs of Internal Brain Y

As mentioned in section II-A, the inputs to Y consist of two parts, one from X (bottom-up) and the other from Z (top-down).

The neurons in AL have the local receptive fields from X area (input image) shown as Fig. 3. Suppose the receptive field is $a \times a$, the neuron (i, j) in AL perceives the region $R(x, y)$ in the input image ($i \leq x \leq (i+a-1)$, $j \leq y \leq (j+a-1)$), where the coordinate (i, j) represents the location of the neuron on the two-dimensional plane shown as Fig. 1

and similarly the coordinate (x, y) denotes the location of the pixel in the input image.

Likewise, the neurons in DL have the global receptive fields from Z area including TM and LM. It is important to note that in Fig. 1, each Y neuron has a limited input field in X but a global input field in Z .

Finally, PL combines the outputs of the above two levels, AL and DL, and output the signals to motor area Z .

B. Release of neurons

After the initialization of the network, all the Y neurons are in the initial state. With the network learning, more and more neurons which are allowed to be turned into the learning state will be released gradually via this biologically plausible mechanism. Whether a neuron is released depends on the status of its neighbor neurons. As long as the release proportion of the region with the neuron at the center is over p_0 , this neuron will be released. In our experiments, the region is $3 \times 3 \times d$ (d denotes the depth of Y area) and $p_0 = 5\%$.

C. Pre-response of the Neurons

It is desirable that each brain area uses the same area function f , which can develop area specific representation and generate area specific responses. Each area A has a weight vector $\mathbf{v} = (\mathbf{v}_b, \mathbf{v}_t)$. Its pre-response value is:

$$r(\mathbf{v}_b, \mathbf{b}, \mathbf{v}_t, \mathbf{t}) = \dot{\mathbf{v}} \cdot \dot{\mathbf{p}} \quad (1)$$

where $\dot{\mathbf{v}}$ is the unit vector of the normalized synaptic vector $\mathbf{v} = (\mathbf{v}_b, \mathbf{v}_t)$, and $\dot{\mathbf{p}}$ is the unit vector of the normalized input vector $\mathbf{p} = (\mathbf{b}, \mathbf{t})$. The inner product measures the degree of match between these two directions of $\dot{\mathbf{v}}$ and $\dot{\mathbf{p}}$, because $r(\mathbf{v}_b, \mathbf{b}, \mathbf{v}_t, \mathbf{t}) = \cos(\theta)$ where θ is the angle between two unit vectors $\dot{\mathbf{v}}$ and $\dot{\mathbf{p}}$. This enables a match between two vectors of different magnitudes. The pre-response value ranges in $[-1, 1]$.

In other words, if regarding the synaptic weight vector as the object feature stored in the neuron, the pre-response measures the similarity between the input signal and the object feature.

The firing of a neuron is determined by the response intensity measured by the pre-response (shown as Equation 1). That is to say, If a neuron becomes a winner through the top-k competition of response intensity, this neuron will fire while all the other neurons are set to zero. In the network training, both motors' firing is imposed by the external teacher. In testing, the network operates in the free-viewing mode if neither is imposed, and in the location-goal mode if LM's firing is imposed, and in the type-goal mode if TM's is imposed. The firing of Y (internal brain) neurons is always autonomous, which is determined only by the competition among them.

D. Two types of neurons

Considering that the learning rate in Hebbian learning (introduced below) is 100% while the retention rate is 0% when the neuron age is 1, we need to enable each neuron to

autonomously search in the input space $\{\dot{\mathbf{p}}\}$ but keep its age (still at 1) until its pre-response value is sufficiently large to indicate that current learned feature vector is meaningful (instead of garbage-like). A garbage-like vector cannot converge to a desirable target based on Hebbian learning.

Therefore, there exist two types of neurons in the Y area (brain) according to their states, initial state neurons (ISN) and learning state neurons (LSN). After the initialization of the network, all the neurons are in the initial state. During the training of the network, neurons may be transformed from initial state into learning state, which is determined by the pre-response of the neurons. In our network, a parameter ϵ_1 is defined. If the pre-response is over $1 - \epsilon_1$, the neuron is transformed into learning state, otherwise, the neuron keeps the current state.

E. Top-k Competition

Top-k competition takes place among the neurons in the same area, imitating the lateral inhibition which effectively suppresses the weakly matched neurons (measured by the pre-responses). Top-k competition guarantees that different neurons detect different features. The response r'_q after top-k competition is

$$r'_q = \begin{cases} (r_q - r_{k+1}) / (r_1 - r_{k+1}) & \text{if } 1 \leq q \leq k \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where r_1 , r_q and r_{k+1} denote the first, q th and $(k + 1)$ th neuron's pre-response respectively after being sorted in descending order. This means that only the top-k responding neurons can fire while all the other neurons are set to zero.

In Y area, due to the two different states of neurons, top-k competition needs to be modified. There exist two kinds of cases:

- If the neuron is ISN and the pre-response is over $1 - \epsilon_1$, it will fire and be transformed into the learning state, otherwise keep the current state (i.e., initial state).
- If the neuron is LSN and the pre-response is over $1 - \epsilon_2$, it will fire.

So the modified top-k competition is described as:

$$r''_q = \begin{cases} r'_q & \text{if } r'_q > \epsilon \\ 0 & \text{otherwise} \end{cases}$$

$$\epsilon = \begin{cases} 1 - \epsilon_1 & \text{if neuron is ISN} \\ 1 - \epsilon_2 & \text{if neuron is LSN} \end{cases}$$

where r'_q is the response defined in Equation 2.

F. Hebbian-like Learning

The concept of neuronal age will be described before introducing Hebbian-like learning. Neuronal age represents the firing times of a neuron, i.e., the age of a neuron increases by one every time it fires. Once a neuron fires, it will implement hebbian-like learning and then update its synaptic weights and age. There exist a close relation between the neuronal age and the learning rate. Put simply, a neuron with lower age has higher learning rate and lower retention rate. Just like human, people usually lose some memory capacity

as they get older. At the “birth” time, the age of all the neurons is initialized to 1, indicating 100% learning rate and 0% retention rate.

Hebbian-like learning is described as:

$$\mathbf{v}_j(n) = w_1(n)\mathbf{v}_j(n-1) + w_2(n)r'_j(t)\mathbf{p}_j(t)$$

where $r'_j(t)$ is the response of the neuron j after top-k competition, n is the age of the neuron (related to the firing times of the neuron), $\mathbf{v}_j(n)$ is the synaptic weights vector of the neuron, $\mathbf{p}_j(t)$ is the input patch perceived by the neuron, w_1 and w_2 are two parameters representing retention rate and learning rate with $w_1 + w_2 \equiv 1$. These two parameters are defined as following:

$$w_1(n) = 1 - w_2(n), \quad w_2(n) = \frac{1 + u(n)}{n}$$

where $u(n)$ is the amnesic function:

$$u(n) = \begin{cases} 0 & \text{if } n \leq t_1 \\ c(n - t_1)/(t_2 - t_1) & \text{if } t_1 < n \leq t_2 \\ c + (n - t_2)/r & \text{if } t_2 < n \end{cases}$$

where $t_1 = 20, t_2 = 200, c = 2, r = 10000$ [21].

Only the firing neurons (firing neurons are in learning state definitely) and all the neurons in initial state will implement Hebbian-like learning, updating the synaptic weights according to the above formulas. The age of the neurons in learning state and initial state is updates as

$$n(t+1) = \begin{cases} n(t) & \text{if the neuron is ISN} \\ n(t) + 1 & \text{if the neuron is top-k LSN.} \end{cases}$$

Generally, a neuron with lower age has higher learning rate. That is to say, ISN is more capable to learn new concepts than LSN. If the neurons are regarded as resources, ISNs are the idle resources while LSNs are the developed resources. So, the resources utilization (RU) in Y area can be calculates as

$$RU = \frac{N_{\text{LSN}}}{N_{\text{LSN}} + N_{\text{ISN}}} \times 100\%$$

where RU represents the resources utilization, N_{LSN} and N_{ISN} are the number of LSN and ISN.

G. How each Y neuron matches its two input fields

All Y neurons compete for firing via the above top-k mechanisms. The initial weight vector of each Y neuron is randomly self-assigned, as discussed below. We would like to have all Y neurons to find good vectors in the input space $\{\mathbf{p}\}$. A neuron will fire and update only when its match between $\hat{\mathbf{v}}$ and $\hat{\mathbf{p}}$ is among the top, which means that the match for the bottom-up part $\hat{\mathbf{v}}_b \cdot \hat{\mathbf{b}}$ and the match for the top-down part $\hat{\mathbf{b}}_t \cdot \hat{\mathbf{t}}$ must be both top. Such top matches must be sufficiently often in order for the neuron to mature.

This gives an interesting but extremely important property for attention — relatively very few Y neurons will learn background, since a background patch does not highly correlated with an action in Z .

Whether a sensory feature belongs to a foreground or background is defined by whether there is an action that often co-occurs with it.

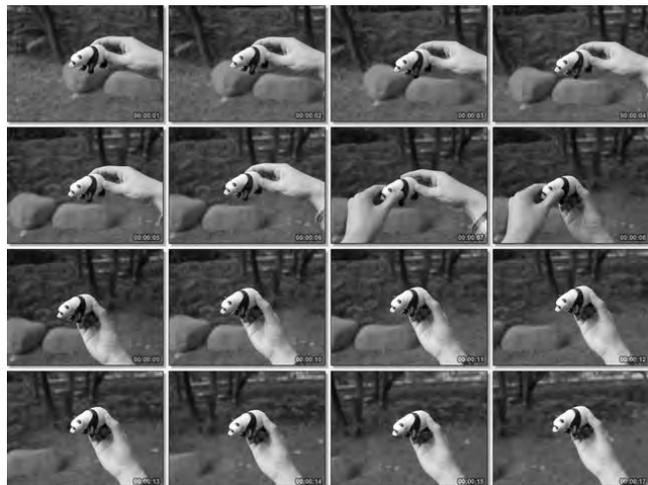


Fig. 5: Frames extracted from a continuous video clip and used in the training and testing of the network

IV. EXPERIMENTS AND RESULTS

A. Sample Frames Preparation from Natural Videos

In our experiment, 10 objects shown in Fig.4 have been learned. The raw video clips of each object to be learned were completely taken in the real natural environments. During video capture, the object held by the teacher’s hand was required to move slowly so that the agent could pay attention to it. Fig. 5 shows the example frames extracted from a continuous video clip as an illustration which needs to be preprocessed before fed into the network. The pre-processing described below is automatically or semi-automatically via hand-crafted programs.

- 1) Resize the image extracted from the video clip to fit the required scales demanded in the network training and testing.
- 2) Provide the correct information including the type, scale and location of the sample in each extracted image with natural backgrounds as the standard of test and the supervision in Z area, just like what the teacher does.

B. Experiment Design

In our experiment, the size of each input image is set to 32×32 for X area. For sub-areas Y_1, Y_2, Y_3 and Y_4 with individual receptive fields $7 \times 7, 11 \times 11, 15 \times 15$ and 19×19 are adopted in Y area. And totally 10 different types of objects (i.e., TM has 10 neurons) with 11 different scales (from 16×16 to 26×26 , i.e., SM has 11 neurons) are used in Z area. For each scale of objects, the possible locations is $(32 - S + 1) \times (32 - S + 1)$ ($S = 16, 17, \dots, 26$), i.e., LM has 17×17 neurons considering that objects with different scales can have the same location. In addition, if the depth of each Y area is 3, the total number of Y neurons is $26 \times 26 \times 3 + 22 \times 22 \times 3 + 18 \times 18 \times 3 + 14 \times 14 \times 3 = 5040$, which can be regarded as the resources of network.

The training set consisted of even frames of 10 different video clips, with one type of foreground object per video.

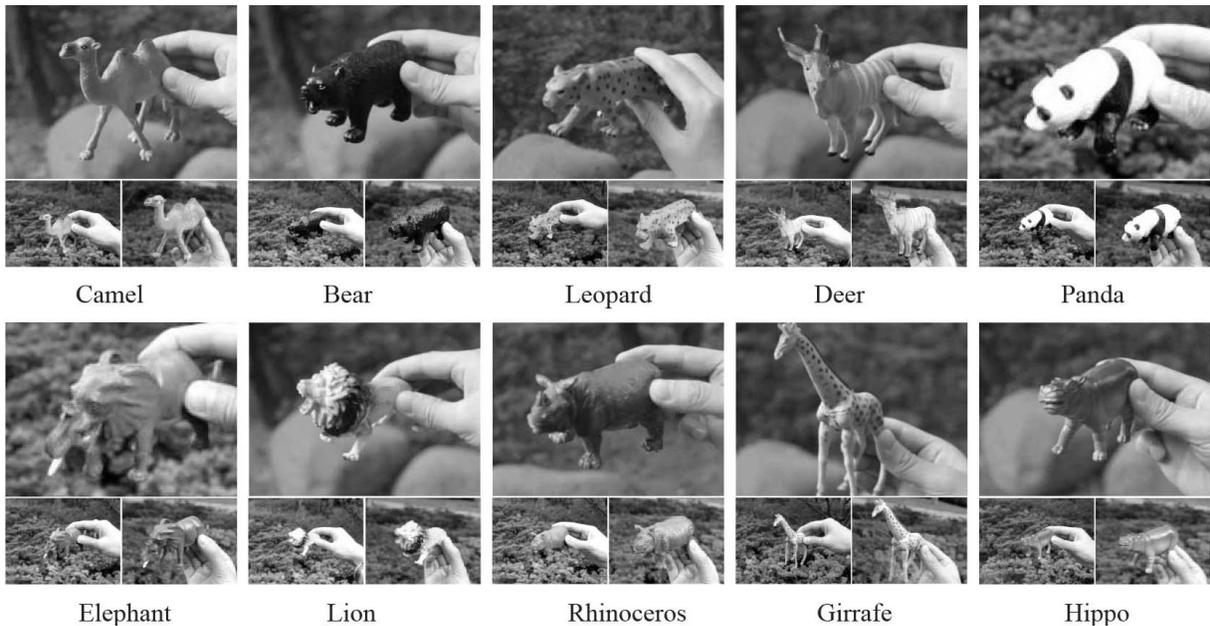


Fig. 4: The pictures on the top visualize 10 objects to be learned in the experiment. The lower-left and the lower-right pictures show the smallest and the largest scale of the objects, respectively (the size of the pictures carries no particular meaning).

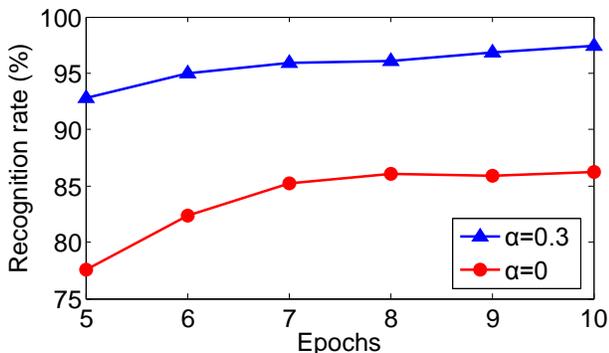


Fig. 6: Recognition rate variation within 6 epochs (from epoch 5th to 10th) under $\alpha = 0$ and $\alpha = 0.3$.

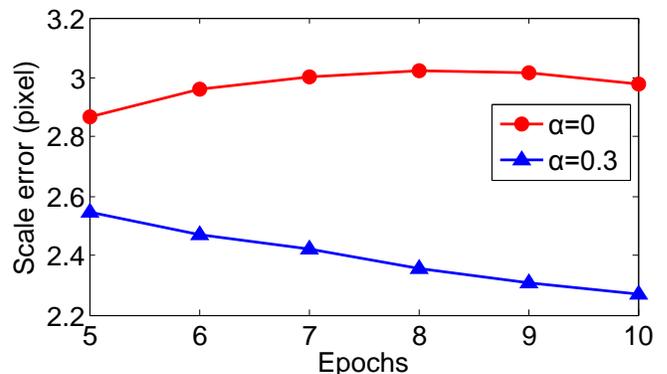


Fig. 7: Scale error variation within 6 epochs (from epoch 5th to 10th) under $\alpha = 0$ and $\alpha = 0.3$.

For each training epoch, every object with every possible scale is learned at every possible location (pixel-specific). So, there are 10 classes $\times (17 \times 17 + 16 \times 16 + 15 \times 15 + 14 \times 14 + 13 \times 13 + 12 \times 12 + 11 \times 11 + 10 \times 10 + 9 \times 9 + 8 \times 8 + 7 \times 7)$ locations = 16940 different training cases and the network is about $1 - 5040/16940 = 70.2\%$ short of resources to memorize all these cases. The test set consisted of odd frames of 10 video clips to guarantee the difference of both foreground and background in the network training phase and testing phase. Multiple epochs are applied to observe the performance modification of the network by testing every foreground object at every possible location after each epoch.

C. Network Performances

The pre-response of Y neurons is calculated as

$$\mathbf{r}(\mathbf{v}_b, \mathbf{b}, \mathbf{v}_t, \mathbf{t}) = (1 - \alpha)\mathbf{r}^b(\mathbf{v}_b, \mathbf{b}) + \alpha\mathbf{r}^t(\mathbf{v}_t, \mathbf{t}) \quad (3)$$

where \mathbf{r}_b is the bottom-up response and \mathbf{r}_t is the top-down response. Parameter α is applied to adjust the coupling ratio of top-down part to bottom-up part in order to control the influence on Y neurons from these two parts. This bottom-up, top-down coupling is not new. The novelty is twofold: first, the top-down activation originates from the previous time step ($t - 1$) and second, non-zero top-down parameter ($\alpha > 0$) is used in the testing phase. These simple modifications create a temporally sensitive network. In formula 3, top-down response \mathbf{r}^t consists of three parts from TM, SM and LM respectively. In our experiments, the percentage of energy for each section is the same, i.e., 1/3.

The high responding Z neurons (including TM, SM and LM) will boost the pre-response of the Y neurons correlated with those neurons more than the other Y neurons mainly correlated with other classes, scales and locations. This can be regarded as top-down biases. These Y neurons' firing

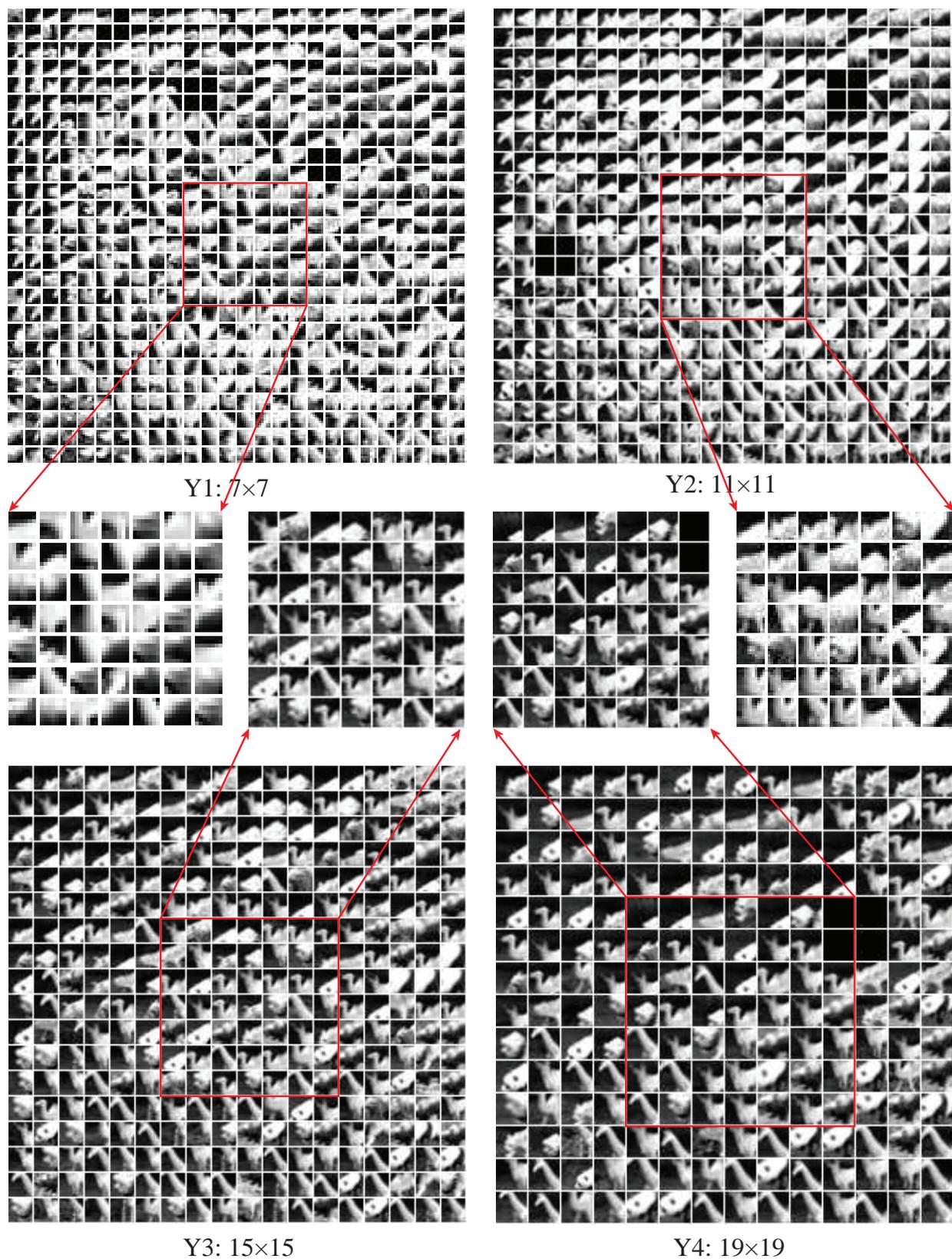


Fig. 9: Visualization of the bottom-up weights of the neurons in the first depth of each Y area. Each small square patch visualized a neuron's bottom-up weights vector, whose size represents the receptive field. The black image patch indicates the corresponding neuron is in the initial state.

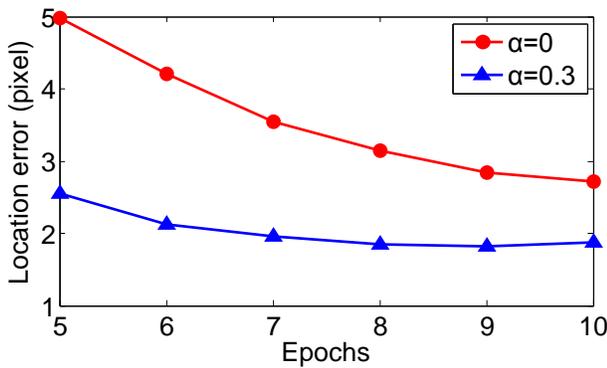


Fig. 8: Location error variation within 6 epochs (from epoch 5th to 10th) under $\alpha = 0$ and $\alpha = 0.3$.

leads to a stronger chance of firing of certain Z neurons without taking into account the actual next image (if $\alpha = 1$). This top-down signal is thus generated regarded as an expectation of the next frame's output. The actual next image also stimulates the corresponding neurons (feature neurons) to fire from the bottom-up. The combined effect of these two parts is controlled by the parameter α . When $\alpha = 1$, the network state ignores subsequent image frames entirely. When $\alpha = 0$, the network operates in a frame-independent way (i.e., free-viewing, not influenced by top-down signal).

The performance of the network, including type recognition rate, scale error and location error, is shown as Fig 6, 7 and 8. In each figure, two performance curves, which corresponds to two conditions, $\alpha = 0$ and $\alpha = 0.3$, are drawn. As discussed above, parameter α controls the ratio of top-down versus the bottom-up part. The higher α is, the stronger the expectations triggered by the top-down signal is. These three figures indicate that the motor initiated expectations through top-down connections have improved the network performance to a certain extent.

In order to investigate the internal representations of WWN-7 after learning the specific objects in the natural video frames, the bottom-up synaptic weights of the neurons in four Y areas with different receptive fields are visualized in Fig 9. Multiple scales of object features are detected by the neurons in different Y areas shown as the figure.

V. CONCLUSION

In this paper, based on the prior work, a new biologically-inspired developmental network WWN-7 has been proposed to deal with general recognition of multiple objects with multiple scales. From the results of experiments, WWN-7 showed its capability of learning multiple concepts (i.e., type, scale and location) concurrently from continuous video taken from natural environments. Besides, in WWN-7, temporal context is used as motor initiated expectation through top-down connections, which has improved the network performances shown in our experiments.

In the future work, more objects with different scales and views will be used in experiment to further verify the performance of WWN-7. And an ongoing work is to study the

influence of the parameter α on the network performance and try to implement the autonomous and dynamical adjustment of the percentage of energy for each section (i.e., bottom-up, TM, SM and LM).

REFERENCES

- [1] J. E. Laird, A. Newell, and P. S. Rosenbloom, "Soar: An architecture for general intelligence," *Artificial Intelligence*, vol. 33, pp. 1–64, 1987.
- [2] J. Weng, "Symbolic models and emergent models: A review," *IEEE Trans. Autonomous Mental Development*, vol. 3, pp. +1–26, 2012, accepted and to appear.
- [3] L. G. Ungerleider and M. Mishkin, "Two cortical visual systems," in *Analysis of visual behavior*, D. J. Ingel, Ed. Cambridge, MA: MIT Press, 1982, pp. 549–586.
- [4] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature Neuroscience*, vol. 2, no. 11, pp. 1019–1025, Nov. 1999.
- [5] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust object recognition with cortex-like mechanisms," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 411–426, 2007.
- [6] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *Journal of Physiology*, vol. 160, no. 1, pp. 107–155, 1962.
- [7] J. Weng, "On developmental mental architectures," *Neurocomputing*, vol. 70, no. 13–15, pp. 2303–2323, 2007.
- [8] J. Weng, "A theory of developmental architecture," in *Proc. 3rd Int'l Conf. on Development and Learning (ICDL 2004)*, La Jolla, California, Oct. 20–22 2004.
- [9] J. Weng, "A 5-chunk developmental brain-mind network model for multiple events in complex backgrounds," in *Proc. Int'l Joint Conf. Neural Networks*, Barcelona, Spain, July 18–23 2010, pp. 1–8.
- [10] Z. Ji, J. Weng, and D. Prokhorov, "Where-what network 1: "Where" and "What" assist each other through top-down connections," in *Proc. IEEE Int'l Conference on Development and Learning*, Monterey, CA, Aug. 9–12 2008, pp. 61–66.
- [11] Z. Ji and J. Weng, "WWN-2: A biologically inspired neural network for concurrent visual attention and recognition," in *Proc. IEEE Int'l Joint Conference on Neural Networks*, Barcelona, Spain, July 18–23 2010, pp. +1–8.
- [12] M. Luciw and J. Weng, "Where-what network 3: Developmental top-down attention for multiple foregrounds and complex backgrounds," in *Proc. IEEE International Joint Conference on Neural Networks*, Barcelona, Spain, July 18–23 2010, pp. 1–8.
- [13] M. Luciw and J. Weng, "Where-what network-4: The effect of multiple internal areas," in *Proc. IEEE International Joint Conference on Neural Networks*, Ann Arbor, MI, Aug 18–21 2010, pp. 311–316.
- [14] X. Song, W. Zhang, and J. Weng, "Where-what network-5: Dealing with scales for objects in complex backgrounds," in *Proc. IEEE International Joint Conference on Neural Networks*, San Jose, California, July 31–Aug 5 2011, pp. 2795–2802.
- [15] Y. Wang, X. Wu, and J. Weng, "Skull-closed autonomous development: Wwn-6 using natural video," in *Proc. IEEE International Joint Conference on Neural Networks*, Brisbane, QLD, 10–15 June 2012, pp. 1–8.
- [16] Y. Wang, X. Wu, and J. Weng, "Skull-closed autonomous development," in *18th International Conference on Neural Information Processing, ICONIP 2011*, Shanghai, China, 2011, pp. 209–216.
- [17] Y. Wang, X. Wu, and J. Weng, "Synapse maintenance in the where-what network," in *Proc. IEEE International Joint Conference on Neural Networks*, San Jose, California, July 31–Aug 5 2011, pp. 2822–2829.
- [18] J. Weng, "Three theorems: Brain-like networks logically reason and optimally generalize," in *Proc. Int'l Joint Conference on Neural Networks*, San Jose, CA, July 31 - August 5 2011, pp. +1–8.
- [19] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen, "Autonomous mental development by robots and animals," *Science*, vol. 291, no. 5504, pp. 599–600, 2001.
- [20] J. Weng, "Why have we passed "neural networks do not abstract well"?" *Natural Intelligence*, vol. 1, no. 1, pp. 13–23, 2011.
- [21] J. Weng and M. Luciw, "Dually optimal neuronal layers: Lobe component analysis," *IEEE Trans. Autonomous Mental Development*, vol. 1, no. 1, pp. 68–85, 2009.

Serious Game Modeling of Caribou Behavior Across Lake Huron using Cultural Algorithms and Influence Maps

James Fogarty

Dep. of Computer Science, Wayne State University
Detroit, MI
Fogarty@valassis.com

Xiangdong Sean Che

Dep. of Computer Science, Eastern Michigan University
Ypsilanti, Michigan, U.S.A
xche@emich.edu

J. O'Shea

Museum of Anthropology, University of Michigan
Ann Arbor, USA
joshea@umich.edu

Robert G. Reynolds

Dep. of Computer Science, Wayne State University
Detroit, MI
Reynolds@cs.wayne.edu

Areej Salaymeh

Dep. of Computer Science, Wayne State University
Detroit, MI
Eg6674@wayne.edu

Abstract—Recent surveys of a stretch of terrain underneath Lake Huron have indicated the presence of a land bridge which would have existed 10,000 years ago, during the recession of ice during the last Ice Age, connecting Canada and the United States. This terrain, dubbed the Alpena-Amberley land bridge, was host to a full tundra environment, including migratory caribou herds. Analysis of the herds, their potential behavior and the likely areas of their movement would lead researchers to the locations Paleo-Indians would pick for hunting and driving the animals

The application designed around these concepts used Microsoft's .Net platform and XNA Framework in order to visually model this behavior and to allow the entities in the application to learn the behavior through successive generations. By utilizing an influence map to manage tactical information, and cultural algorithms to learn from the maps to produce path planning and flocking behavior, paths were discovered and areas of local concentration were isolated. In particular, paths emerged that focused on efficient migratory behavior at the expense of food consumption, which caused some deaths. On the other hand paths emerged that focused on food consumption with only gradual migration process. Then here were also strategies that emerged that blended both goals together; making effective progress towards the goal without excessive losses to starvation.

Keywords—Cultural Algorithms, social fabric, virtual world models, path planning, influence maps, learning group movement)

I. INTRODUCTION

Computer modeling of group behavior and ecological modeling has seen considerable development as shown with Walter and Bergman [1][2], but there remains work to be done to integrate the two together for qualitative results [1][2]. Previous research has focused on either discovering the ecological basis for behavior in both modeling the terrain and

flora [3], by modeling the herbivore movements in relative isolation to environments [2], or by modeling the individual aspects of herbivore movements without analyzing group behavior as a whole [1].

We choose to construct a virtual world model of an ancient environment, the Alpena-Amberley land bridge [6][10]. This project initiated by Dr. John O'Shea, a University of Michigan anthropologist, was undertaken to better understand how prehistoric Paleo-Indians hunted and lived 10,000 years ago. At that time, the level of what is now modern Lake Huron was low enough to expose a 6 mile wide land bridge that connected what is today Alpena in Michigan to the Amberley area in Canada. The land bridge is now submerged beneath 200 feet of water. O'Shea speculated that it contained evidence of prehistoric occupation. A preliminary sonar survey of selected areas on the land bridge supported by an NSF High Risk research grant provided evidence to support this conjecture. The data collection activities were performed using sonar, autonomous underwater vehicles, and scuba divers. The preliminary results offered tempting insight into what could have existed 10,000 years ago. This resulted in the project being named as one of the top 100 scientific discoveries of 2009 by Discover Magazine [35].

Reynolds and a group of students in the Artificial Intelligence Laboratory at Wayne State University began investigating the possibility of recreating a virtual world model of the region, a model that can be used by the archaeologists to predict where to do further surveys and investigations. Since the overall area was very large and surveys, both above and under the water, are costly, initial simulations of the region were small in scale involving small numbers of animals and hunters over a limited region, but returned promising results

[4][29]. Here we elect to expand upon this earlier work to create a large scale serious game.

A serious game is a game designed for a purpose other than entertainment, but rather with a main purpose of training and investigating. It will utilize a detailed world in which group behavioral concepts are ascertained and the best and most likely scenarios of life in this arctic world will rise to the top. Since Cultural Algorithms, developed by Reynolds, are particularly adept at the process of modeling societies [17] [34], we will use them to design human and animal group behavior in these extended models. Cultural Algorithms are a branch of evolutionary computation that model the cultural evolution process. A Cultural Algorithm consists of a belief space and a population space that communicate through an interaction protocol whereby the belief space influences the population and the best individuals can in turn influence the belief space [37] [38]. This process is based on acceptance and influence functions and thus the population evolves according to the promotion of the best individuals' beliefs.

Figure 1 below displays at an abstract level the Cultural algorithm process. The population is initialized in the first step labeled, "Population" Each individual is scored against objective criteria and a predetermined number of elite, those with the best performances, are selected to update the Belief Space. This Belief Space is the foundation for the genetic makeup of the offspring for succeeding generations. The process is then repeated and over much iteration, the population converges on results, which are applicable to the problem space.

Our stated goal is to simulate the emergence of likely caribou behavior, positioning and survival across the Alpena-Amberley ridge during various scenarios that are supported by and designed with real world flora and fauna constraints. By utilizing both real world terrain data acquired from Dr. O'Shea as well as simulated human behavior that Cultural Algorithms (CA) and influence maps will help develop, we hope to create representations of what actual events transpired on this land.

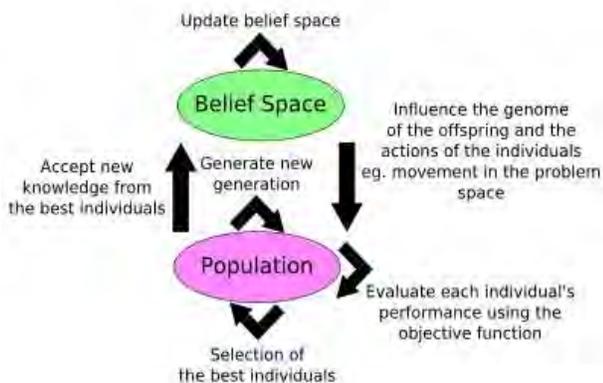


Fig. 1. Design of Cultural Algorithms.

This paper describes an influence map driven, Cultural Algorithm that generates path-planning for caribou migration routes across the Alpena-Amberley land bridge. The results will be displayed not only in a real-time 3D display of migration behavior, but also as well as a 2D output of influence values.

A. Terrain Creation and Modeling

Using underwater depth, latitude and longitude positioning information provided by Dr. John O'Shea [4][6][37], the geo-positional data was used to construct a grey-scale image called a height-map. This height-map is the basis for all of our results. Using Microsoft's XNA Framework, we will generate a 3D model mesh [7] and with refinements, allow access to height and normal data throughout the simulation.

The land bridge itself extended from Alpena, Michigan, USA, to Amberley, Ontario, Canada during the last ice age, and is pictured in figure 2 [37]. It was a strip of land that crossed under what is currently Lake Huron. The research has already provided some insight into possible hunting and camping sites. The process of using sonar to map the lake bottom has given researchers the ability to construct a 3D interactive and expandable environment for behavioral and cultural analysis using constructs that would help discover the possible survival processes of these societies that relied on this terrain. Current survey work being conducted by Dr. O'Shea promises to yield far higher resolution scans of the underwater terrain. The simulation has been developed to incorporate changes in configurations of terrain and vegetation coverage, as well as variables such as water level. The latter becomes important in the future when modeling the impact of rising water levels on land bridge utilization.

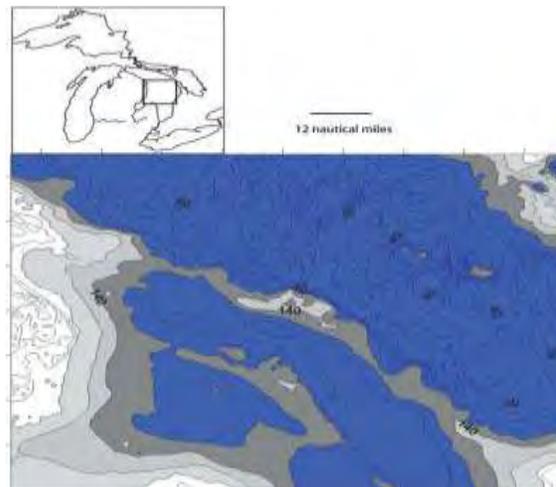


Fig. 2. Alpena-Amberley bridge across Lake Huron

B. Group Behavior

There are a large number of available path planning methods for individuals and while they may be extended to maintain groups and formations [36]. On the other hand, if we abstract the population into discrete sets of individuals and then plan paths according to the abstract group entity instead of the individual inside, we can achieve both accurate, low-overhead

path planning as well as visibly fluid movement. To prevent both the overhead and to tightly integrate behavior to simulate a real work flocking behavior, we look at the behavior introduced by Reynolds [5].

Shown below in Figure 3 is an example of the execution of our path-finding technique which is described in this paper. The dark squares are path-finding nodes which are determined by a heuristic detailed in section II – each square is an in-order node that the group will navigate towards. However, since the groups are comprised of many individuals with their own personal actions, they will simulate fluid behavior on the way. Their individual actions are determined by the parameters, such as desired proximity to neighbors. This behavior allows groups to have macro goal decision making ability while also allowing individuals to make choices.

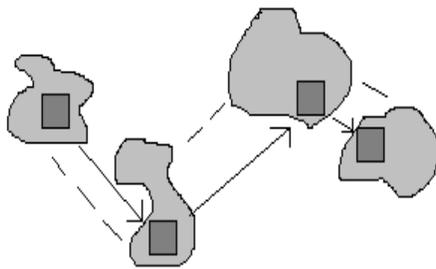


Fig. 3. Rigid path-finding nodes shown as dark blocks, individual's movement of groups around the nodes while traveling shown in lighter colored areas.

Entities create a dynamic flocking movement using three primary principles: cohesion, separation, and alignment. Using these three forces, a flow is established that takes into account the individuals in each group. Splitting apart the total population into groups of dynamic sizes with capacities and orientation centers, as well as establishing individual goal locations and weights for each group, allows multiple interactions with the environment with each run. These goals and weights are created and tracked through the first portion of our learning mechanisms, influence maps. The refinement of each portion is done at the end of each generational pass using the Cultural Algorithm, which applies the data for the next generations input influence maps, path-finding and weights.

C. Learning Mechanisms

Our process involves a two-step learning method. The first step is to generate influence maps, an assigned-value, cellular method of summarizing values in space. Influence maps in games and simulations have a strong history [32] and were found to be of great use in maintaining a statistical tracking of the interactive world in computer games. Influence maps are cellular divisions of 2D or 3D worlds with tactical values assigned to the spaces they occupy. This value is determined by a problem specific function and can be accessed during the application execution for both value input into the program, or to receive output values for later reference by learning and decision making algorithms. The map is a function of the game

world, distributed across the terrain which represents the desirability of a particular cell, reflecting positive influences such as availability of food and negative influences, such as dangerous elevation changes. An influence map is generated by subdividing the world space into smaller segments or cells; these segments can be directly accessed by a number of methods that allow an entity in the world to retrieve informational values concerning the area they require information about.

In our program, the influence maps are generated in real time during the run of the serious game. By using influence maps, we can track in aggregate valuable components of tactical knowledge that would influence caribou behavior: location of food, rough terrain, or any other positive or negative influence to a segment of 3D space. The influence maps are input into the simulation by using grey-scale bitmaps, allowing a starting point of tactical knowledge; the particular setup can be saved out at any time as well.

The second step, and most refined, involves the use of Cultural Algorithms. CA has proven itself as an adaptable source for both real time and turn based game applications [30] [31] and since we use both real-time components as well as those which occur at the end of each belief cycle, we should consider whether to focus on the real-time or turn-based advantages of Cultural Algorithms. The Open Racing Car Simulator (TORCS) [31] is closest to what we are developing – a real-time 3D virtual simulation with reward and punishment concepts (win/loss), which we can extrapolate to our own world. The CA in TORCS had access to state variables for a vehicle such as gearing, track position, wheel spin, and fuel; the CA would take that information and optimize the output to interact with the vehicle. The TORCS system interface allowed that refined information to generate a set of output parameters to interact with the vehicle, such as acceleration and braking, that could move the car within the race pack. Likewise, behavior researched by Vitale [29] used a CA to evaluate a simple wandering kinematic in the same situation we now attempt to model. A key parameter in the kinematic was the jittering parameters that would control the direction and rate of change, the groups of individuals modeled became more successful at crossing the land bridge. The Cultural Algorithm in that game simulation was used to parameterize a kinematic wandering algorithm that is applied to all caribou agents in order to determine how much random movement the caribou undergo. Figure 4 shows a number of individuals denoted as triangles as they attempt to cross a land bridge in a simulation using the wandering kinematic. Each candidate set of parameters in the Cultural Algorithm population was evaluated in terms of its ability to effectively control group behavior. The performance was evaluated in terms of distance traveled and percentage of living entities at the end of a run, allowing the CA to construct well defined parameters for crossing a sample bridge as it improves over generations.

The 3D virtual world is developed with the ability to extend the algorithm to multiple animals and with a variety of inputs, simply by changing the value files the program reads. However, the implementation itself sets out to discover valid and hopefully verifiable results on caribou migration over the bridge based that can be used to interpret existing real world

data from underwater surveys. The remainder of this paper is organized as follows. Section II provides the details of the group learning framework as a whole and how it integrates all the various behaviors into one final product. It will discuss in detail how the Influence Map driven Cultural Algorithm framework will attempt to produce through tandem behavior an optimal result for our expectations; that is, a predicted and reasonable path over the land bridge. Section III reviews the performance of the framework relative to our goals. It will present and dissect the final product and determine the applicability to real world situations and applications. It will also discuss the ease and flexibility of future extensions to the framework and its components. Section IV summarizes our findings and presents directions for future work.

II. IMPLEMENTATION

Development for the application was executed in Microsoft's C# language on top of the .Net platform. The supporting structure of the graphical components was built around Microsoft's XNA 3.1 Framework, a .Net series of classes allowing for rapid development of graphical software for display and interaction on multiple .Net compatible machines. The graphics are rendered on the GPU and all game logic and AI procedures are executed on the CPU. The application was designed with both scalability and extensibility in mind.

A. Terrain and Entity Display and Interaction

The terrain display process begins with converting a 2D grey-scale height-map into 3D mesh model for display on the GPU through the XNA Framework as shown in Figure 5. A custom content processor is used instead of the normal content processing in XNA.



Fig. 4. Improved caribou movement over the land bridge

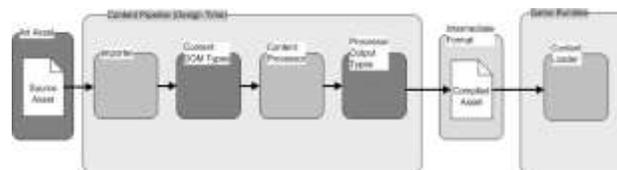


Fig. 5. Microsoft's XNA Content Pipeline

There are two important variables that must be specified in order to coerce the pixels to a vector in 3-dimensional space: height and spacing. The height dictates the maximum range in display units between black and white input pixels; the spacing dictates the XZ (horizontal) spacing between adjacent pixels in the input file.

With that information, each pixel is converted to a point in 3D space originating at $[0, 0, 0]$ and growing in positive directions – each point becomes a vertex in a final 3D mesh for display. Textures are mapped to this model mesh based on local vertex heights and the normals of the faces from 3 adjacent vertices. The terrain during game time has a useful class called HeightMapInfo which allows the height and normal of the terrain at any point in space of the simulation to be read and ingested into the game logic.

Vegetation is read from 2 grey-scale maps: one generated for trees and one for scrub brush. The vegetation is displayed in a manner called “bill-boarding”, where a 2D image is rendered in 3D space and in this case, always faces the camera. All visible vegetation features – such as trees and scrub – are drawn using bill-boarding techniques as described by Pettit [33]. In bill-boarding, a 2D image is rendered as a 3D point with surface area. In this scenario, the images always rotate in the XZ axes while maintaining the Y up vector, to simulate forests and loads of bushes. The input image locations are scaled in the same XZ manner as the terrain and the height for the bill-boarded images is determined by the terrain height selected with their XZ location. Multiple billboards can be placed in close proximity if the intensity on the 2D input image is higher – this random clustering creates a realistic looking environment. The vegetation is constructed as a single mesh for more efficient drawing. There are also two 3D meshes behind the scenes which duplicate the display of the terrain, which is used for determining the maximum density of vegetation at any 3D point, in the same way height is read from the terrain.

The water is displayed using a highly realistic method generated from [7] and incorporates reflections, refractions and other high-fidelity options to give a tactile feel to the game world.

All game entities are displayed on the GPU using built in classes provided by XNA, which allowed for rapid development. The entities also inherit from the XNA class called DrawableGameComponent, which is a class which has overrideable functions for updating and drawing them, allowing time elapsed during each step of the execution to be referenced for smooth movement. During each time step, before being drawn, an entity goes through an update phase.

During this period and on the entity level, items such as caloric count and living status are updated. This information is used by other processes, especially flocking and CA components.

B. Influence Map

The single class InfluenceMap supports all of the behavior for creating and updating the influence map across the game. When created, a map has the specified number of cells and the cells have system-determined dimensions so that all cell's dimensions are equal and the XYZ sizes are proportionate to the terrain dimensions and the number of cells specified. The map can be instantiated in one of two different ways:

Providing the terrain dimensions and the number of cells desired in XYZ directions. The map is constructed by taking the terrain dimensions and dividing that by the number of cells for height, width and length. The cells all have an initial value of 0. A 2D grey-scale image is provided. The size of the image in pixels determines the number of cells in the X and Z directions and a single Y level is assigned. The values for the cells are determined by the grey-scale values of the pixels, from 0 – 255.

The class supports a number of functions, such as updating a cell's value by index or by position, as well as finding the lowest or highest value cells and the ability to save the resultant influence map as a bitmap image. This process is shown visually in Figure 6.

When assigning values to an influence map for a particular run, a number of items are considered:

a). Availability of food: the vegetation maps are scanned. In this example, the two underlying vegetation maps are dissected to determine density values, which exist between 0 and 255. Scrub vegetation has a higher relative positive weight than do the tree values, due to more accessible food and less of an impediment to traversal (acts in tandem with #3). Therefore, our influence map for vegetation values is based on the formula $(t + (1.5 * s))$, where t is the value of tree coverage from [0, 255] in that cell and s is the value of scrub coverage in that cell from [0, 255].

b). Dangers previous generations have encountered. When the influence map is calculated at the beginning of a generation, the deaths the previous generation encountered are strong negatives in a particular cell and as such reduce the likelihood of a path passing through there. For each caribou that dies in a cell in any previous generation, 0.5 a point of a [0, 255] range is deducted permanently from that cell.

c). Terrain difficulty. Since the difficulty of passing through dense trees is accounted for by reducing the food value of trees in #1, we look at elevation change in a particular cell. We do not consider rapidly changing values such as jagged locations, but only the maximum undulation. By making higher resolution cells, we can emulate the tracking of jagged materials. The height variation between the center of a cell and it's neighbor reduces that cell value by $1/h$, where h is the maximum height of the terrain. In other words, a step of 100 units in terrain with maximum elevation of 1000 is 0.1 reduction in that cell's value.

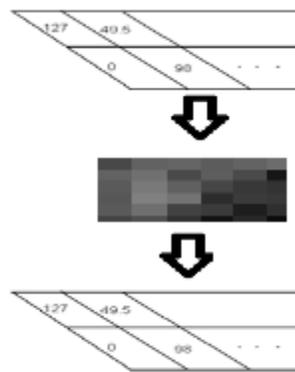


Fig. 6. Converting an influence map from values to image and back

d). Proximity to the final goal: the influence map will be seeded with a final “arrival” location which will signal the completion of the trek. On arrival within that cell, the herd is considered done for that generation.

Those values represent our initial construction of the influence map. As food is eaten, the cell containing caribou is reduced by 1.0 for each caribou present in each cell. Therefore, when selecting nodes for traversal, nodes which are already supporting large numbers of caribou will not be selected. This map is reset at the end of each generation. A second persistent map is used to track caribou deaths over each generation.

C. Flocking

The flocking behavior borrows heavily from Reynold’s research in [5] and Microsoft’s Game Development Library in [26], with a few key differences. The flock behavior is defined as not only the interaction between animals, but also a weighted and overarching goal of following a sequential list of goal behavior. Base flocking implementation is based on the three movement constraints of cohesion, separation and alignment; each caribou is assigned to a herd at the start and all of the herds start at half capacity. Caribou re-assignment during runtime is important to the final evaluation of success and is calculated as follows:

```
If (Dist(herd[i].center,entity[x])> Dist(herd[j].center, entity[x])
    && herd[j].count < herd[j].capacity)
    herd[i].remove(caribou[x])
    herd[j].add(caribou[x])
```

Goals are managed by using a simple list. At the end of every herd behavior update cycle, the goal list is checked. If the list is not empty, the weight for this particular herd is applied to the final heading and each caribou adjusts according to that weight. The weight is limited within certain bounds to prevent unnatural behavior, as excessively heavy weighting will cause erratic movement.

The weighting also has a specific modification added to it to encourage the herd to actually transit the land bridge, as opposed to allowing them to engage in purely appetite or safety driven behavior. This weight will be calculated at two different

points – a value affecting where the next path node is selected by shifting the location further towards the opposing side of the land bridge and as a variable in the Cultural Algorithm genome which will be carried and modified through generations as a deciding factor in how much value should be placed on completion – if too low, caribou will not cross the bridge in time, if too high, they will pass through areas without enough food and subsequently die.

D. Path-finding

Path-finding uses the A* algorithm to determine the path based on weights and distances derived from the influence maps; the path the A* algorithm determines directly corresponds to the goal list of the particular herd requesting a path.

The A* implementation creates a series of nodes based on the center of each cell in the influence map. From this point, the weight of the connecting edges is determined by the change in cell value; for negative changes, since A* is a nonnegative directed graph, the value is simply set to 0. An example route is shown in figure 7:

7	6	5	6	7	8	9	10	11		19	20	21	22
6	5	4	5	6	7	8	9	10		18	19	20	21
5	4	3	4	5	6	7	8	9		17	18	19	20
4	3	2	3	4	5	6	7	8		16	17	18	19
3	2	1	2	3	4	5	6	7		15	16	17	18
2	1	0	1	2	3	4	5	6		14	15	16	17
3	2	1	2	3	4	5	6	7		13	14	15	16
4	3	2	3	4	5	6	7	8		12	13	14	15
5	4	3	4	5	6	7	8	9	10	11	12	13	14
6	5	4	5	6	7	8	9	10	11	12	13	14	15

Fig. 7. Sample A* path-finding route – grey are obstacles, green is start, blue is finish and the numbers inside each cell are the distance so far plus the weight of the heuristic (in this case, distance)

The principle of the method is based on the current distance traveled plus the estimated distance left to reach the goal; the estimated distance is heuristic driven. The path with the estimated shortest distance is tried until it is proven that it is longer than the next shortest estimated distance or the goal is reached

Our goal is simple: we wish to maximize the number of arriving caribou across our terrain with each passing cycle until we reach our threshold. With this in mind, we take our design for our individuals, which are herds, and their chromosomes which shall be maximized, which are defined in Table 1.

Our first generation is initialized with parameters within the following range. These values will represent our normative knowledge in the Cultural Algorithm, as they represent the allowable range for an individual in this population:

TABLE 1 Initialization Values

Parameter	Type	Value
herdSize	Int	capacity / 2
detectionDist	Float	40.0f– 00.0f
separationDist	Float	30.0f– 70.0f
oldDirInfluence	Float	0.5f– 1.5f
flockDirInfluence	Float	0.5f– 1.5f
randomDirInfluence	Float	0.01f– 0.1f
perMemberWeight	Float	0.5f– 1.5f
finalGoalWeight	Float	0.0f– 1.0f

We select a common herd size and capacity for all herds in order to prevent certain arriving herds from being unfairly favored or discriminated against due to a poor initial size. For similar reasons, our locations are all initialized within a smaller starting box, allowing intermingling of herds at the very beginning. This has the added bonus of being a real world simulation of funneling herds into the land bridge as they move from a larger area to smaller. All herds likewise share the same initial goals, since a common influence map is shared between the CA's individuals (herds). We initiate all movement within the box on the left below, and an individual herd is considered “arrived” if the calculated center arrives within the box to the right.

We have mentioned that our genome represents the normative knowledge of our Belief Space. The rest of our Cultural Algorithm's knowledge is derived from influence maps. This vegetation – and the intensity to which our individuals seek it – is an important source of individual success. Also, we track caribou deaths via a separate influence map, which is treated as situational and temporal knowledge – caribou that have expired in particular areas at particular times are used as a behavior variable in achieving success of crossing the land-bridge.

Once all individuals have arrived, or at least one has arrived after the time allotted has expired, the real-time component ends, and the CA begins to compare the surviving individuals. The individuals are ranked on the following objective function:

$$v = \text{herdCount} * \text{avgHerdCalories} * (1 / (\text{MaxTime} / \text{herdTransTime})) \quad (1)$$

avgHerdCalories is the average calorie count and is normalized between 1 and the maximum herd capacity, to prevent small herds who have fed extremely well from tilting the results in their favor. This function is useful in many aspects. Herds with the largest number of surviving members are rewarded the most, as are those with high calorie counts. We also consider the inverse of the travel time to be a valuable factor here, as we do not have a fully real-time system capable of changing seasons and we consider this to be during fall and spring migration, where a short transit time is valuable.

Influence maps, representing several sources of knowledge, are updated each time a caribou expires with the negative information associated with the death. The top 10% of surviving performers are elected to update the normative knowledge genome. Based on their scores above, an average successful gene is created and merged with the current best values that already exist in our Belief Space – this current best value is shifted 50% of the way towards what we have just elected as the best average genome.

This newly updated Belief Space is now communicated to the individual population based on our selected topological operator, which in this case is fully-connected: every individual receives some influence on their behavior, in the necessary direction according to the values described in table 2. This has the effect of moving individuals towards those who have already proven themselves to be successful, which is now reflected in the Belief Space.

TABLE 2 Transition Values Based on Success

Parameter	Type	Value
herdSize	Int	capacity / 2
detectionDist	Float	+/- 0.5f
separationDist	Float	+/- 0.5f
oldDirInfluence	Float	+/- 0.01f
flockDirInfluence	Float	+/- 0.01f
randomDirInfluence	Float	+/- 0.002f
perMemberWeight	Float	+/- 0.01f
finalGoalWeight	Float	+/- 0.01f

Herd size is reset to half of capacity and the next generation runs until we reach our terminating condition.

III. EXPERIMENTAL FRAMEWORK AND RESULTS

In order to demonstrate the suitability of the mechanisms selection chosen for this application, we construct and run 10 simulations and dissect the results statistically to determine their validity. Each simulation was run on a stable machine and the results will be detailed in the section below. The system

TABLE 4 Starting Values for Test Runs

Parameter	Run									
	1	2	3	4	5	6	7	8	9	10
herdSize	50	50	50	50	50	50	50	50	50	50
detectionDist	47.3	65.1	92.5	56.8	80.2	51.4	93.7	76.2	85.6	95.8
separationDist	69.0	45.8	62.2	43.0	64.8	50.1	62.1	52.3	68.9	51.9
oldDir	0.7	0.6	1.1	0.8	1.1	0.9	1.1	1.1	1.2	1.2
flockDir	1.4	0.7	1.3	0.9	1.4	1.3	0.5	1.1	1.1	0.8
randomDir	0.06	0.04	0.05	0.09	0.09	0.01	0.04	0.08	0.03	0.02
perMemberWeight	1.0	1.4	1.1	0.6	1.2	0.6	1.0	0.7	0.8	1.3
finalGoalWeight	0.8	0.0	0.96	0.85	0.61	0.71	0.89	0.01	0.67	0.89

specifications are described in Table 3:

TABLE 3 Test Platform Specifications

CPU	Intel x86 Dual 3.2GHz
RAM	4GB DDR2 1066
GPU	1GB Palit 9600 GT Sonic

A. Parameter Design

All of our experiments were executed on a single generated terrain height-map, tree map, and scrub map. The maps were all grey scale images generated and imported by the methods defined in previous section. These topographical images are the 2D representations of the 3D land bridge across which the caribou will be migrating.

With the terrain information above, 10 separate runs of 100 generations a piece were made with the starting parameters described in table 4. Runs 1 through 5 were generated running south to north, and runs 6 through 10 were running north to south. All the weights are randomly initiated within specific ranges. Five herds were used for each run simultaneously. Presented below are the initialization values of the herd that would be selected as the most influential herd of the first generation; their final results – the last generation – are shown in table 5. The details of exemplary runs will be shown afterwards. When viewing the results, the white areas on the grayscale maps are traversable and the black areas are not.

We will first present our run results and then discuss what they mean. In Table 5 the results of the most successful herd of the final generation is displayed. The pertinent weights for group behavior are shown, as well as the sizes of the finishing herds. Further analysis of select runs can be found below the table. The “Avg Nutrition” is based on a range from 0-100 and that distance traveled is the most accurate distance between waypoints based on estimates of terrain and data synchronicity.

TABLE 5 Results of Best Herd for Final Generation for Each Run

parameter	Run									
	1	2	3	4	5	6	7	8	9	10
Starting Size	50	50	50	50	50	50	50	50	50	50
Finishing Size	44	36	41	39	41	42	50	34	47	48
Starvation Count	6	14	9	11	9	8	0	16	3	2
Avg Nutrition	73	89	65	21	72	16	25	71	89	21
Dist Traveled	11710	6581	8423	16715	12451	8189	8634	7731	11314	13459
detectionDist	54.8	54.3	87.1	63.3	90.7	44.9	100.0	41.7	100.0	96.3
separationDist	61.0	37.5	53.7	51.5	45.3	61.3	65.6	53.3	72.4	50.4
oldDir	0.51	0.5	1.35	0.92	0.85	1.44	1.41	0.61	1.19	1.1
flockDir	1.41	1.5	0.81	1.05	1.35	1.12	1.01	0.95	0.86	1.5
randomDir	0.06	0.048	0.06	0.1	0.18	0.032	0.028	0.1	0.054	0.06
perMemberWeight	0.78	1.0	0.82	0.2	0.90	0.45	0.93	0.46	0.51	0.69
finalGoalWeight	0.762	0.510	0.932	0.81	0.642	0.941	0.812	0.6	0.814	0.932

We will now discuss several notable runs to show how certain runs are successful or unsuccessful. Figure 8 below shows the direction the caribou are traveling across our digital land bridge as tied to the actual bathymetric data.

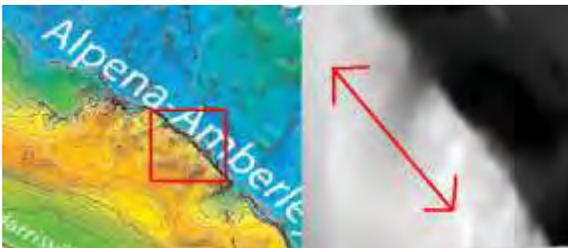


Fig. 8. The Direction and Location of our simulation

Figure 9 below shows our survival rate over the 10 runs. Notice the steadily increasing values for all individuals, indicating a continuous improvement based on changing Belief Space values. The Y axis is the number of surviving herd members, the X axis is the generation.

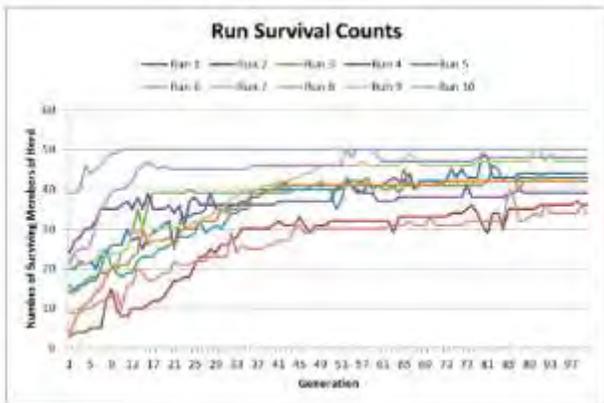


Fig. 9. The run comparisons

Runs #7 and #9 were the most successful runs, with some of the highest survival rates at the end of the 100 requisite runs; the migratory pattern carried the caribou from South to North. By comparing relative values we see that a middle-to-high finalGoalWeight, which directs the herd towards the destination, did not overwhelm the search for food, allowing a greater number of caribou to consume enough and survive the transit. We also see that a high detection distance allowed us to have a higher selection range of cells to choose for our path nodes. This path combined both the grazing and migration goals together.

Runs #2 and #8, running North to South and South to North respectively, shared a common problem with a very low starting parameter to seek the other end of the land-bridge. Therefore the herds ended up in high food areas, they did not starve, except they grazed until the allocated time began to run out and they drifted through areas already consumed and would starve. These paths represented more of a grazing approach than a migration pathway. With a higher goal-seeking desire, they would traverse the bridge faster, not having to continue to wander for food. However, as the belief space was updated, more and more would begin to cross, though they were never as successful as the others within this generation limit.

Runs #7 and #10 was another example of a successful run which went from North to South. Due to a high finalGoal variable, the caribou were able to guarantee that they would migrate across the land bridge in time; also, their high detection allowed them to plot a transitory path which would lead them through certain areas of high nutritional value. However, their very high goal weights made them cross with little respect for food, leading to low nutrition, even with a high survival rate.

By comparing our run results, we find that there are a few important variables to consider. We cannot seek the goal too strongly, or we will miss locations which have a higher proportion of food and thus will begin to starve out the herd. We also cannot have too high a wandering variable, as we will spend too long transitioning the terrain, just as a high old

direction drive will keep caribou trapped in particular locations or vectors for too long. A high detection distance, which is utilized in both the ability to plan paths and find other members of the herd to follow, is also useful in locating efficient pathways by giving a broader spectrum of possible choices. Therefore we need to blend or balance the two goals in order to get a migration pathway that does not take too long but on the other hand does not lose too many individuals. A high success rate, being based on timely completion and well fed caribou, should have high detection distance, a moderate migration goal and a moderate random component for food discovery.

Below, in figures 10 and 11, we can see run time examples of caribou behavior across the land bridge, showing their grazing and migratory actions.



Fig 10: Caribou grazing



Fig 11: Caribou Movement

IV. CONCLUSION

In this paper we have developed an approach to learning migratory behavior using Cultural Algorithms. The Cultural Algorithms generate and use influence maps to compute path-planning behavior using the A* algorithm. The resultant runs shown the different degrees to which the goal of migration and food procurement can be combined. That is, if one moves too quickly, some will not be able to get sufficient amount so food and will die along the way. On the other hand, if one focuses on food procurement then migration behavior can slow to a stand still. This might be behavior exhibited by the caribou during the summer season.

To produce a solid and sustained migration one needs to balance the food procurement goal and the directional goals together. These behaviors emerged from the system as a result of the cultural learning process. What remains to be done is to integrate in a defensive component in herd migratory behavior relative to human and other predators. That is a topic for future work discussed below.

A. Future Work

We have seen that the serious game modeling of caribou migration using Cultural Algorithms to learn successful migratory behavior has had success in generating plausible behavior; there are a number of aspects which are open for further and more refined development. Many of these components can be integrated with the simulation easily and with external development only; an example would be more precise terrain generation – to integrate the new data, only new maps would need to be supplied.

Paleolithic hunters often constructed drive lanes, cairns and campsites when canvassing the land bridge. While the current construction allows the inclusion of an influence map to simulate the existence of such materials in affecting animal behavior, further work would allow specifications and implementation according to archeological records.

Weather and seasons would be an appropriate addition, allowing the simulation to display multiple variations that would affect the transition across the land bridge.

Other fauna would complete the ecosphere with the addition of food competitors and hunters of the caribou. For example, particular types of herbivores in the tundra environment may compete directly with caribou and should be modeled as such; wolves in hostile environs also would have an effect on behavior and generation growth and transition.

More dynamic influence maps would allow for shifting priorities when creating the influence maps which are the guiding influence over the bridge and the main source of input towards the path selection, which in many ways determines how the herds move.

Creating influence maps which will take into effect changing seasons and different foliage encourage the maps to be trained against any discovered real world data. Modifications to influence map values during run time is currently supported, but logic for animal consumption, environmental effects and others should be included to change the terrain as the herds drive through. This will also need to include changes to the display parameters for visual cues.

REFERENCE

- [1] Walters, Carl J., Hilborn, R., and Peterman, R. Computer Simulation of Barren-Ground Caribou Dynamics. *Ecological Modeling*, 1 (1975), 303-315.
- [2] Bergman, Carita M., Schaefer, J., and Luttich, S. Caribou Movement as a Correlated Random Walk. *Oecologia*, Vol 123, 3 (2000), 364-274.
- [3] Bliss, L. C., Courtin, G. M., Pattie, D. L., Riewe, R. R., Whitefield, D. W. A., and Widden, P. Arctic Tundra Ecosystems. *Annual Review of Ecology and Systematics*, Vol 4 (1973), 359-399.
- [4] Reynolds, R. G., O’Shea, J., Fotouhi, F., Fogarty, J., Vitale, K., Meadows, G. R. *Agile Design of Reality Games Online*. Supported in part by NSF High-Risk Grant #BCS-0829324.

- [5] Reynolds, Craig W. Flocks, Herds, and Schools: A Distributed Behavioral Model. *Computer Graphics*, Vol 21, 4 (1987).
- [6] O'Shea, J. Ancient Hunters and the Lake Stanley Causeway: A Pilot Study. NSF High-Risk Grant #BCS-0829324, University of Michigan, Ann Arbor. (2008).
- [7] Lobao, A. S., Evangelista, B. P., Leal de Farias, J. A., Grootjans, R. Beginning XNA 3.0 Game Programming: From Novice to Professional, Apress, 1 Edition (2009).
- [8] Jain, A. Fundamentals of Digital Image Processing, Prentice-Hall, 1986, Chapter 7.
- [9] Griffin, L. D. Proceedings: Mathematical, Physical and Engineering Sciences. Vol. 456, No. 2004 (Dec. 8, 2000), pp. 2995-3004.
- [10] Lewis, C. F. M., Heil Jr., C. W., Hubeny, J. B., King, J. W., Moore Jr., T. C., Rea, D. K. The Stanley unconformity in Lake Huron Basin, *Journal of Paleolimnology*, Vol. 9, 2004.
- [11] Cowling, S. A., Sykes, M. T., Bradshaw, R. H. W. Palaeovegetation-Model Comparisons, Climate Change and Tree Succession in Scandinavia over the past 1500 year, *Journal of Ecology*, Vol. 89, No. 2 (Apr., 2001), pp. 227-236
- [12] Saranoha, D. A. Mathematical Models of Tundra Communities and Populations. Laboratory of Mathematical Ecology, Computer Center, USSR Acad. Sci, Vavilova 40, 117333 Moscow USSR.
- [13] Price, D. T., Halliwell, D. H., Apps, M. J., Peng, C. H. Adapting a Path Model to Simulate the Sensitivity of Central-Canadian Boreal Ecosystems to Climate Variability, *Journal of Biogeography*, Vol. 26, No. 5 (Sep., 1999), pp. 1101-1113.
- [14] Sirois, L., Bonan, G. B., Shugart, H. H. Development of a Simulation Model of the Forest-Tundra Transition Zone of Northeastern Canada, *Can. J. For. Res.* Vol. 24 (4), pp. 697-706 (1994).
- [15] Benedict, J. B. Tundra Game Drives: An Arctic-Alpine Comparison, *Arctic, Antarctic, and Alpine Research*, Vol. 37, No. 4 (Nov., 2005), pp. 425-434).
- [16] Reynolds, R. G., An Introduction to Cultural Algorithms, in [EO94], pp. 131-139.
- [17] Reynolds, R. G., Peng, B., Whallon, R. Emergent Social Structures in Cultural Algorithms.
- [18] Reynolds, R. G., Saleem, S. M. The impact of environmental dynamics on cultural emergence. In *Perspectives on Adaptations in Natural and Artificial Systems*. Oxford University Press, 2001.
- [19] Stout, B. Smart Moves: Intelligent Pathfinding. *Game Developer Magazine*, July, 1997.
- [20] Lester, P. A* Pathfinding for Beginners. July, 2005.
- [21] Jin, X., Reynolds, R. G. Using Knowledge-based evolutionary computation to solve nonlinear constraint optimization problems: a cultural algorithm approach. *Proceedings of the 1999 Congress on Evolutionary Computation (1999)*, 1672-1678.
- [22] Haupt, R. L., Haupt, S. E. *Practical Genetic Algorithms*. Wiley-Interscience, 2nd Edition, 1998.
- [23] Reynolds, R. G., Peng, B. Knowledge learning and social swarms in cultural algorithms. *Journal of Mathematical Sociology*, 29 (2005), 1-18.
- [24] Chung, C. J., Reynolds, R. G. A testbed for solving optimization problems using cultural algorithms, *Evolutionary Programming*, 1996, pp. 225-236.
- [25] Ali, M., Reynolds, R. G. Embedding a social fabric component into cultural algorithms toolkit for an enhanced knowledge-drive engineering optimization. *International Journal of Intelligent Computing and Cybernetics*, 1, 4 (2008), 563-597.
- [26] Microsoft XNA Education Catalogue. June 30th, 2010. <http://creators.xna.com/en-US/education/>
- [27] Reynolds, R. G., Ali, M., and Jayyousi, T. Mining the social fabric of archaic urban centers with cultural algorithms. *Computer*, 41, 1 (January, 2008), pp. 64-72.
- [28] Best, Christopher. Multi-Objective Cultural Algorithms. Master Thesis, Wayne State University, Detroit. (2009)
- [29] Vitale, Kevin. Learning Group Behavior in Games Using Cultural Algorithms and the Land Bridge Simulation Example. Master Thesis, Wayne State University, Detroit. (2009)
- [30] Ochoa, A., Padilla, A., Gonzalez, S., Castro, A., and Hal, S. Improve a Game Board based on Cultural Algorithms. *The International Journal of Virtual Reality*, 2008, 7(2). pp. 41-46.
- [31] Loiacono, D., Togelius, J., Lanzi, P. L., Kinnaird-Heether, L., Lucas, S. M., Simmerson, M., Perez, D., Reynolds, R. G., and Saez, Y. The WCCI 2008 Simulated Car Racing Competition.
- [32] Tozour, P. "Influence Mapping," *Game Programming Gems 2*, 2001.
- [33] Pettit, C. J., Sheth, F., Harvey, W., and Cox, M. *Building a 3D Object Library for Visualizing Landscape Futures*. 18th World IMACS/MODSIM Congress, 2009.
- [34] Srinivasan, S. and Ramakrishnan, S. "Cultural Algorithm Toolkit for Multi-Objective Rule Mining," *International Journal on Computational Sciences & Applications (IJCSA)*, vol. 2, no. 4, pp. 9-23, Aug. 2012.
- [35] Barth, Amy. "Top 100 Stories of 2009." *Discover Magazine* December 17th, 2009, Accessed October 22nd 2010 <http://discovermagazine.com/2010/jan-feb/095>
- [36] McInnes, C. R. *Velocity field path-planning for single and multiple unmanned aerial vehicles*. *The Aeronautical Journal*, July 2003, pp 419-426.
- [37] Jin, J. "Path Planning in Reality Games Using Cultural Algorithm: The Land Bridge Example." M.S. thesis, Wayne State University, Detroit, Michigan, 2011.
- [38] Liu, D. "Multi-Objective Cultural Algorithms." Doctor of Philosophy Thesis, Wayne State University, Detroit, Michigan, 2011.

Establish the Three Theorems: DP Optimally Self-Programs Logics Directly from Physics

Juyang Weng

Abstract—In artificial intelligence (AI) there are two major schools, symbolic and connectionist. The Developmental Program (DP) self-programs logic into a Developmental Network (DN) directly from physics or data. Weng 2011 [6] proposed three theorems about the DN which bridged the two schools: (1) From any complex FA that demonstrates human knowledge through its sequence of the symbolic inputs-outputs, the DP incrementally develops a corresponding DN through the image codes of the symbolic inputs-outputs of the FA. The DN learning from the FA is incremental, immediate and error-free. (2) After learning the FA, if the DN freezes its learning but runs, it generalizes optimally for infinitely many image inputs and actions based on the embedded inner-product distance, state equivalence, and the principle of maximum likelihood. (3) After learning the FA, if the DN continues to learn and run, it “thinks” optimally in the sense of maximum likelihood based on its past experience. This paper presents the proofs.

I. INTRODUCTION

The major differences between a symbolic network (SN) and a Developmental Network (DN) are illustrated in Fig. 1.

Marvin Minsky 1991 [4] and others argued that symbolic models are logical and clean, while connectionist (he meant emergent) models are analogical and scruffy. The logic capabilities of emergent networks are still unclear, categorically.

Computationally, feed-forward connections serve to feed sensory features [5] to motor area for generating behaviors. It has been reported that feed-backward connections can serve as class supervision [2], attention [1], and storage of time information.

In the following, we analyze how the DN theory bridges the symbolic school and the connectionist school.

II. DP ALGORITHM

The small DP algorithm self-programs logic into a huge DN directly from physics. A DN has its area Y as a “bridge” for its two banks, X and Z . If Y is meant for modeling the entire brain, X consists of all receptors and Z consists of all muscles neurons. Y potentially can also model any Brodmann area in the brain. The most basic function of an area Y seems to be prediction — predict the signals in its two vast banks X and Z through space and time.

Algorithm 1 (DP): Input areas: X and Z . Output areas: X and Z . The dimension and representation of X and Y areas are hand designed based on the sensors and effectors of the

Juyang Weng is a professor at the Department of Computer Science and Engineering, Cognitive Science Program, and Neuroscience Program, Michigan State University, East Lansing, MI, 48824 USA (email: weng@cse.msu.edu) and a visiting professor at the School of Computer Science and Engineering at Fudan University, Shanghai, China.

The author would like to thank Mr. Hao Ye at Fudan University who carefully proof read the proofs presented here.

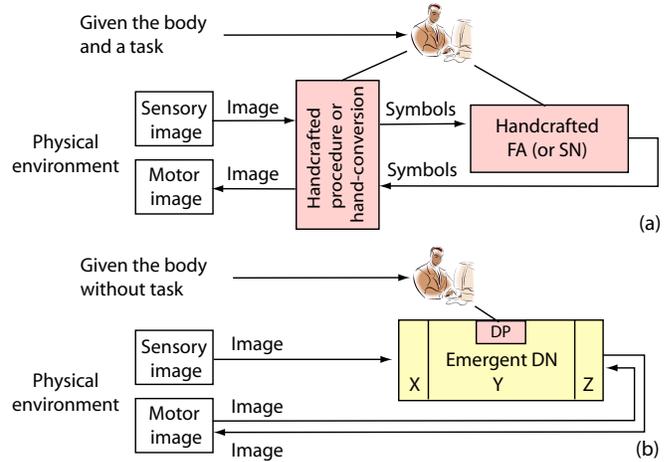


Fig. 1. Why a symbolic machine (or Symbolic Network SN, probabilistic or non-probabilistic) is task-specific and cannot self-program but the DP for an emergent automaton DN is not task-specific and can self-program DN for a wide variety of tasks directly from physics. An SN uses handcrafted task-specific, handcrafted representations, IA, OA, and IOM. For example, the human programmer perceives the relevant patch and denotes it as a symbol σ and disregards other parts $\bar{\sigma}$ in the current input image. The DP of a DN is not task-specific, so that the DP self-programs logic directly from the physical world. It attends a patch in image \mathbf{x} that corresponds to σ and attends part of \mathbf{z} to predict the next X image \mathbf{x}' and the next Z image \mathbf{z}' . An effector image \mathbf{z} is equivalent to the corresponding state q which includes all the currently attended context in space and time (perception and action). All the pink blocks are handcrafted. All the yellow blocks are emergent.

species (or from evolution in biology). Y is the skull-closed (inside the brain), not directly accessible by the outside.

- 1) At time $t = 0$, for each area A in $\{X, Y, Z\}$, initialize its adaptive part $N = (V, G)$ and the response vector \mathbf{r} , where V contains all the synaptic weight vectors and G stores all the neuronal ages. For example, use the generative DN method discussed below.
- 2) At time $t = 1, 2, \dots$, for each A in $\{X, Y, Z\}$ repeat:
 - a) Every area A performs mitosis-equivalent if it is needed, using its bottom-up and top-down inputs \mathbf{b} and \mathbf{t} , respectively.
 - b) Every area A computes its area function f , described below,

$$(\mathbf{r}', N') = f(\mathbf{b}, \mathbf{t}, N)$$

where \mathbf{r}' is its response vector.

- c) For every area A in $\{X, Y, Z\}$, A replaces: $N \leftarrow N'$ and $\mathbf{r} \leftarrow \mathbf{r}'$.

III. FORMULATIONS

In the remaining discussion, we assume that Y models the entire brain. If X is a sensory area, $\mathbf{x} \in X$ is always supervised. The $\mathbf{z} \in Z$ is supervised only when the teacher chooses to. Otherwise, \mathbf{z} gives (predicts) motor output.

The area function f which is based on the theory of Lobe Component Analysis (LCA) [7], a model for self-organization by a neural area. Each area A has a weight vector $\mathbf{v} = (\mathbf{v}_b, \mathbf{v}_t)$. Its pre-response vector is:

$$r(\mathbf{v}_b, \mathbf{b}, \mathbf{v}_t, \mathbf{t}) = \frac{\mathbf{v}_b}{\|\mathbf{v}_b\|} \cdot \frac{\mathbf{b}}{\|\mathbf{b}\|} + \frac{\mathbf{v}_t}{\|\mathbf{v}_t\|} \cdot \frac{\mathbf{t}}{\|\mathbf{t}\|} = \dot{\mathbf{v}} \cdot \dot{\mathbf{p}} \quad (1)$$

which measures the degree of match between the directions of $\dot{\mathbf{v}} = (\mathbf{v}_b/\|\mathbf{v}_b\|, \mathbf{v}_t/\|\mathbf{v}_t\|)$ and $\dot{\mathbf{p}} = (\mathbf{b}/\|\mathbf{b}\|, \mathbf{t}/\|\mathbf{t}\|)$.

To simulate lateral inhibitions (winner-take-all) within each area A , top k winners fire. Considering $k = 1$, the winner neuron j is identified by:

$$j = \arg \max_{1 \leq i \leq c} r(\mathbf{v}_{bi}, \mathbf{b}, \mathbf{v}_{ti}, \mathbf{t}). \quad (2)$$

The area dynamically scale top- k winners so that the top- k respond with values in $(0, 1]$. For $k = 1$, only the single winner fires with response value $y_j = 1$ and all other neurons in A do not fire. The response value y_j approximates the probability for $\dot{\mathbf{p}}$ to fall into the Voronoi region of its $\dot{\mathbf{v}}_j$ where the “nearness” is $r(\mathbf{v}_b, \mathbf{b}, \mathbf{v}_t, \mathbf{t})$.

All the connections in a DN are learned incrementally based on Hebbian learning — cofiring of the pre-synaptic activity $\dot{\mathbf{p}}$ and the post-synaptic activity y of the firing neuron. If the pre-synaptic end and the post-synaptic end fire together, the synaptic vector of the neuron has a synapse gain $y\dot{\mathbf{p}}$. Other non-firing neurons do not modify their memory. When a neuron j fires, its firing age is incremented $n_j \leftarrow n_j + 1$ and then its synapse vector is updated by a Hebbian-like mechanism:

$$\mathbf{v}_j \leftarrow w_1(n_j)\mathbf{v}_j + w_2(n_j)y_j\dot{\mathbf{p}} \quad (3)$$

where $w_2(n_j)$ is the learning rate depending on the firing age (counts) n_j of the neuron j and $w_1(n_j)$ is the retention rate with $w_1(n_j) + w_2(n_j) \equiv 1$. The simplest version of $w_2(n_j)$ is $w_2(n_j) = 1/n_j$ which corresponds to:

$$\mathbf{v}_j^{(i)} = \frac{i-1}{i}\mathbf{v}_j^{(i-1)} + \frac{1}{i}\mathbf{1}\dot{\mathbf{p}}(t_i), i = 1, 2, \dots, n_j, \quad (4)$$

where t_i is the firing time of the post-synaptic neuron j . The above is the recursive way of computing the batch average:

$$\mathbf{v}_j^{(n_j)} = \frac{1}{n_j} \sum_{i=1}^{n_j} \dot{\mathbf{p}}(t_i) \quad (5)$$

The initial condition is as follows. The smallest n_j in Eq. (3) is 1 since $n_j = 0$ after initialization. When $n_j = 1$, \mathbf{v}_j on the right side is used for pre-response competition but does not affect \mathbf{v}_j on the left side since $w_1(1) = 1 - 1 = 0$.

A component in the gain vector $y_j\dot{\mathbf{p}}$ is zero if the corresponding component in $\dot{\mathbf{p}}$ is zero.

As we need a slight deviation from the standard definition of FA, let us look at the standard definition first.

Definition 1 (Language acceptor FA): A finite automaton (FA) M is a 5-tuple $M = (Q, \Sigma, q_0, \delta, A)$, where Q is a finite set of states, consists of symbols. Σ is a finite alphabet of input symbols. $q_0 \in Q$ is the initial state. $A \subset Q$ is the set of accepting states. $\delta : Q \times \Sigma \mapsto Q$ is the state transition function.

This classical definition is for a language acceptor, which accepts all strings x from the alphabet Σ that belongs to a language L . It has been proved [3] that given any *regular language* L from alphabet Σ , there is an FA that accepts L , meaning that it accepts exactly all $\mathbf{x} \in L$ but no other string not in L . Conversely, given any FA taking alphabet Σ , the language L that the FA accepts is a regular language. However, a language FA, just like any other automata, only deals syntax not semantics. The semantics is primary for understanding a language and the syntax is secondary.

We need to extend the definition of FA for agents that run at discrete times, as follows:

Definition 2 (Agent FA): A finite automaton (FA) M for a finite symbolic world is a 4-tuple $M = (Q, \Sigma, q_0, \delta)$, where Σ and q_0 are the same as above and Q is a finite set of states, where each state $q \in Q$ is a symbol, corresponding to a set of concepts. The agent runs through discrete times $t = 1, 2, \dots$, starting from state $q(t) = q_0$ at $t = 0$. At each time $t-1$, it reads input $\sigma(t-1) \in \Sigma$ and transits from state $q(t-1)$ to $q(t) = \delta(q(t-1), \sigma(t-1))$, and outputs $q(t)$ at time t , illustrated as $q(t-1) \xrightarrow{\sigma(t-1)} q(t)$.

The inputs to an FA are symbolic. The input space is denoted as $\Sigma = \{\sigma_1, \sigma_2, \dots, \sigma_l\}$, which can be a discretized version of a continuous space of input. In sentence recognition, the FA reads one word at a time. The number l is equal to the number of all possible words — the size of the vocabulary. For a computer game agent, l is equal to the total number of different percepts.

The outputs (actions) from a language acceptor FA are also symbolic, $A = \{a_1, a_2, \dots, a_n\}$ which can also be a discretized version of a continuous space of output. For a sentence detector represented by an FA, when the FA reaches the last state, its action reports that the sentence has been detected.

An agent FA is an extension from the corresponding language FA, in the sense that it outputs the state, not only the acceptance property of the state. The meanings of each state, which are handcrafted by the human programmer but are not part of the formal FA definition, are only in the mind of the human programmer. Such meanings can indicate that a state is an accepting state or not, as a special case of many other meanings associated with the state. However, such concepts are only in the mind of the human system designer, not something that the FA is “aware” of. This is a fundamental limitation of all symbolic models. The Developmental Network (DN) described below do not use any symbols, but instead (image) vectors from the real-world

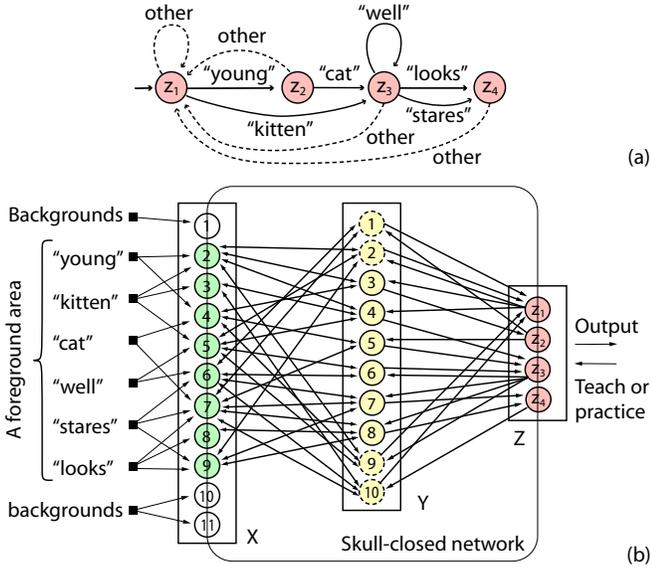


Fig. 2. Conceptual correspondence between an Finite Automaton (FA) with the corresponding DN. (a) An FA, handcrafted and static. (b) A corresponding DN that simulates the FA. It was taught to produce the same input-out relations as the FA in (a). A symbol (e.g., z_2) in (a) corresponds to an image (e.g., $(z_1, z_2, \dots, z_4) = (0, 1, 0, 0)$) in (b).

sensors and real-world effectors. As illustrated in Fig. 2, a DN is grounded in the physical environment but an FA is not.

Fig. 3 gives an example of the agent FA. Each state is associated with a number of cognitive states and actions, shown as text in the lower part of Fig. 3, reporting action for cognition plus a motor action. The example in Fig. 3 shows that an agent FA can be very general, simulating an animal in a micro, symbolic world. The meanings of each state in the lower part of Fig. 3 are handcrafted by, and only in the mind of, the human designer. These meanings are not a part of the FA definition and are not accessible by the machine that simulates the FA.

Without loss of generality, we can consider that an agent FA simply outputs its current state at any time, since the state is uniquely linked to a pair of the cognition set and the action set, at least in the mind of human designer.

A. Completeness of FA

It has been proved [3] that an FA with n states partitions all the strings in Σ into n sets. Each set is called equivalence class, consisting of strings that are indistinguishable by the FA. Since these strings are indistinguishable, any string x in the same set can be used to denote the equivalent class, denoted as $[x]$. Let Λ denote an empty string. Consider Fig. 3. The FA partitions all possible strings into 6 equivalent classes. $[\Lambda] = [\text{"calculus"}]$ as the agent does not know about "calculus" although it is in Σ . All the strings in the equivalent class $[\Lambda]$ end in z_1 . All strings in the equivalent class $[\text{"kitten"} \text{"looks"}]$ end in z_4 , etc.

The completeness of agent FA can be described as fol-

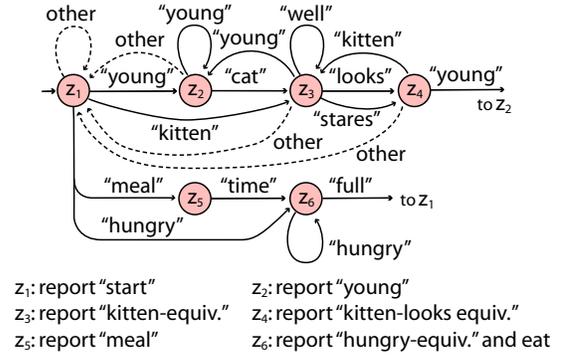


Fig. 3. An FA simulates an animal. Each circle indicates a context state. The system starts from state z_1 . Supposing the system is at state q and receives a symbol σ and the next state should be q' , the diagram has an arrow denoted as $q \xrightarrow{\sigma} q'$. A label "other" means any symbol other than those marked from the out-going state. Each state corresponds to a set of actions, indicated below the FA. The "other" transitions from the lower part are omitted for brevity.

lows. Given a vocabulary Σ representing the elements of a symbolic world, a natural language L is defined in terms of Σ where the meanings of all sentences in L are defined by the set of equivalent classes, denoted by Q . When the number of states is sufficiently large, a properly designed FA can sufficiently characterize the cognition and behaviors of an agent living in the symbolic world of vocabulary Σ .

B. DN Simulates FA

Next, let us consider how a DN learns to simulate any FA. First we consider the mapping from symbolic sets Σ and Q , to vector spaces X and Z .

Definition 3 (Symbol-to-vector mapping): A symbol-to-vector mapping m is a one-to-one mapping $m : \Sigma \mapsto X$. We say that $\sigma \in \Sigma$ and $\mathbf{x} \in X$ are equivalent, denoted as $\sigma \equiv \mathbf{x}$, if $\mathbf{x} = m(\sigma)$.

A binary vector of dimension d is such that all its components are either 0 or 1. It simulates that each neuron, among d neurons, either fires with a spike ($s(t) = 1$) or without ($s(t) = 0$) at each sampled discrete time $t = t_i$. From discrete spikes $s(t) \in \{0, 1\}$, the real valued firing rate at time t can be estimated by $v(t) = \sum_{t-T < t_i \leq t} s(t_i) / T$, where T is the temporal size for averaging. A biological neuron can fire at a maximum rate around $v = 120$ spikes per second, producible only under a laboratory environment. If the brain is sampled at frequency $f = 1000\text{Hz}$, we consider the unit time length to be $1/f = 1/1000$ second. The timing of each spike is precise up to $1/f$ second at the sampling rate f , not just an estimated firing rate v , which depends on the temporal size T (e.g., $T = 0.5\text{s}$). Therefore, a firing-rate neuronal model is less temporally precise than a spiking neuronal model. The latter, which DN adopts, is more precise for fast sensorimotor changes.

Let B_p^d denote the d -dimensional vector space which contains all the binary vectors each of which has at most p components to be 1. Let $E_p^d \subset B_p^d$ contains all the binary vectors each of which has exactly p components to be 1.

Definition 4 (Binary- p mapping): Let $Q = \{q_i \mid i = 1, 2, \dots, n\}$. A symbol-to-vector mapping $m : Q \mapsto B_p^d$ is a binary- p mapping if m is one to one: That is, if $\mathbf{z}_i \equiv m(q_i)$ then $q_i \neq q_j$ implies $\mathbf{z}_i \neq \mathbf{z}_j$.

The larger the p the more symbols the space of Z can represent. However, through a binary- p mapping, each symbol q_i always has a unique vector $\mathbf{z} \in Z$. Note that different q 's are mapped to not only different \mathbf{z} 's but also different directions of \mathbf{z} 's as the input \mathbf{p} of DN is a unit $\hat{\mathbf{p}}$.

Suppose that a DN is taught by supervising binary- p codes at its exposed areas, X and Z . When the motor area Z is free, the DN performs, but the output from Z is not always exact due to (a) the DN outputs in real numbers instead of discrete symbols and (b) there are errors in any computer or biological system. The following binary conditioning can prevent error accumulation, which the brain seems to use through spikes.

Definition 5 (Binary conditioning): For any vector from $\mathbf{z} = (z_1, z_2, \dots, z_d)$, the binary conditioning of \mathbf{z} forces every real-valued component z_i to be 1 if the pre-response of z_i is larger than the machine zero — a small positive bound estimating computer round-off noise.

The output layer Z that uses binary- p mapping must use the binary conditioning, instead of top- k competition with a fixed k , as the number of firing neurons ranges from 1 to p .

C. DP for Generative DN (GDN)

Algorithm 2 (DP for GDN): A GDN is a DN that gives the following specific way of initialization. It starts from pre-specified dimensions for the X and Z areas, respectively. X represents receptors and is totally determined by the current input. But it incrementally generates neurons in Y from an empty Y . Each neuron in Z is initialized by a synaptic vector \mathbf{v} of dimension 0, age 0. Suppose $V = \{\mathbf{v}_i = (\mathbf{x}_i, \mathbf{z}_i) \mid \mathbf{x} \in X, \mathbf{z} \in Z, i = 1, 2, \dots, c\}$ is the current synaptic vectors in Y . Whenever the network takes an input $\mathbf{p} = (\mathbf{x}, \mathbf{z})$, compute the pre-responses in Y . If the top-1 winner in Y has a pre-response lower than 2 (i.e., $\mathbf{p} \notin V$), simulate mitosis-equivalent by doing the following:

- 1) Increment the number of neurons, $c \leftarrow c + 1$.
- 2) Add a new Y neuron. Set the weight vector $\mathbf{v} = \hat{\mathbf{p}}$, its age to be 0, and its pre-response to be 2 since it is the perfect match based on Eq. (1). There is no need to recompute the pre-responses.

The response value of each Z neuron is determined by the starting state (e.g., background class). As soon as the first Y neuron is generated, every Z neuron will add the first dimension in its synaptic vector in the following DN update. This way, the dimension of its weight vector continuously increases together with the number c of Y neurons.

Lemma 1 (Properties of a GDN): Suppose a GDN simulates any given FA using top-1 competition for Y , binary- p mapping, and binary conditioning for Z , and update at least twice in each unit time. Each input $\mathbf{x}(t-1)$ is retained during all DN updates in $(t-1, t]$. Such a GDN has the following properties for $t = 1, 2, \dots$:

- 1) The winner Y neuron matches perfectly with input $\mathbf{p}(t-1) \equiv (q(t-1), \sigma(t-1))$ with $\mathbf{v} = \hat{\mathbf{p}}$ and fires, illustrated in Fig. 4(a) as a single transition edge (red).
- 2) All the synaptic vectors in Y are unit and they never change once initialized, for all times up to t . They only advance their firing ages. The number of Y neurons c is exactly the number of learned state transitions up to time t .
- 3) Suppose that the weight vector \mathbf{v} of each Z neuron is $\mathbf{v} = (p_1, p_2, \dots, p_{c(Y)})$, and Z area uses the learning rate straight recursive average $w_2(n_j) = 1/n_j$. Then the weight p_j from the j -th Y neuron to each Z neuron is

$$\begin{aligned} p_j &= \text{Prob}(j\text{-th } Y \text{ neuron fires} \mid \text{the } Z \text{ neuron fires}) \\ &= f_j/n, \end{aligned} \quad (6)$$

$j = 1, 2, \dots, c(Y)$, where f_j is the number of times the j -th Y neuron has fired conditioned on that the Z neuron fires, and n is the total number of times the Z neuron has fired.

- 4) Suppose that the FA makes transition $q(t-1) \xrightarrow{\sigma(t-1)} q(t)$, as illustrated in Fig. 4(a). After the 2nd DN update, Z outputs $\mathbf{z}(t) \equiv q(t)$, as long as Z of DN is supervised for the 2nd DN update when the transition is received by Z the first time. Z then retains the values automatically till the end of the first DN update after t .

Proof: The proof below is a constructive proof, instead of an existence one. To facilitate understanding, the main ideas are illustrated in Fig. 4. Let the X of the DN take the equivalent inputs from Σ using a symbol-to-vector mapping. Let Z be supervised as the equivalent states in Q , using a binary- p mapping. The number of firing neurons Z depends on the binary- p mapping. The DN lives in the simulated sensorimotor world $X \times Z$ determined by the sensory symbol-to-vector mapping: $m_x : \Sigma \mapsto X$ and the binary- p symbol-to-vector mapping $m_z : Q \mapsto Z$.

We prove it using induction on integer t .

Basis: When $t = 0$, set the output $\mathbf{z}(0) \equiv q(0) = q_0$ for the DN. Y has no neuron. Z neurons have no synaptic weights. All the neuronal ages are zeros. The properties 1, 2, 3 and 4 are trivially true for $t = 0$.

Hypothesis: We hypothesize that the above four properties are true up to integer time t . In the following, we prove that the above properties are true for $t + 1$.

Induction step: During t to $t + 1$, suppose that the FA makes transition $q(t) \xrightarrow{\sigma(t)} q(t+1)$. The DN must do the equivalent, as shown below.

At the next DN update, there are two cases for Y : Case 1: the transition is observed by the DN as the first time. Case 2: the DN has observed the transition.

Case 1: new Y input. First consider Y . As the input $\mathbf{p}(t) = (\mathbf{x}(t), \mathbf{z}(t))$ to Y is the first time, $\hat{\mathbf{p}} \notin V$. Y initializes a new neuron whose weight vector is initialized as $\mathbf{v}_j = \hat{\mathbf{p}}(t)$ and age $n_j = 0$. The number of Y neurons c is incremented by 1 as this is a newly observed state

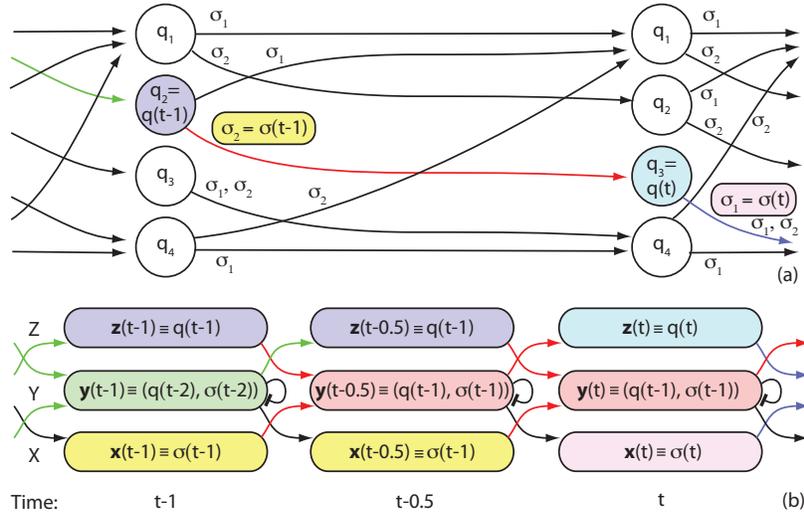


Fig. 4. Model the brain mapping, DN, and SN. In general, the brain performs external mapping $b(t) : X(t-1) \times Z(t-1) \mapsto X(t) \times Z(t)$ on the fly. (a) An NS samples the vector space Z using symbolic set Q and X using Σ , to compute symbolic mapping $Q(t-1) \times \Sigma(t-1) \mapsto Q(t)$. This example has four states $Q = \{q_1, q_2, q_3, q_4\}$, with two input symbols $\Sigma = \{\sigma_1, \sigma_2\}$. Two conditions (q, σ) (e.g., $q = q_2$ and $\sigma = \sigma_2$) identify the active outgoing arrow (e.g., red). $q_3 = \delta(q_2, \sigma_2)$ is the target state pointed to by the (red) arrow. (b) The grounded DN generates the internal brain area Y as a bridge, its bi-directional connections with its two banks X and Z , the inner-product distance, and adaptation, to realize the external brain mapping. It performs at least two network updates during each unit time. To show how the DN learns a SN, the colors between (a) and (b) match. The sign \equiv means “image code for”. In (b), the two red paths from $q(t-1)$ and $\sigma(t-1)$ show the condition $(z(t-1), x(t-1)) \equiv (q(t-1), \sigma(t-1))$. At $t-0.5$, they link to $y(t-0.5)$ as internal representation, corresponding to the identification of the outgoing arrow (red) in (a) but a DN does not have any internal representation. At time t , $z(t) \equiv q(t) = \delta(q(t-1), \sigma(t-1))$ predicts the action. But the DN uses internal $y(t-0.5)$ to predict both state $z(t)$ and input $x(t)$. The same color between two neighboring horizontal boxes in (b) shows the retention of (q, σ) image in (a) within each unit time, but the retention should be replaced by temporal sampling in general. The black arrows in (b) are for predicting X . Each arrow link in (b) represents many connections. When it is shown by a non-black color, the color indicates the corresponding transition in (a). Each arrow link represents excitatory connections. Each bar link is inhibitory, representing top- k competition among Y neurons.

transition. From the hypothesis, all previous Y neurons in V are still their originally initialized unit vectors. Thus, the newly initialized \mathbf{v}_j is the only Y neuron that matches $\dot{\mathbf{p}}(t)$ exactly. With $k = 1$, this new Y neuron is the unique winner and it fires with $y_j = 1$. Its Hebbian learning gives age advance $n_j \leftarrow n_j + 1 = 0 + 1 = 1$ and Eq.(3) leads to

$$\begin{aligned} \mathbf{v}_j &\leftarrow w_1(n_j)\dot{\mathbf{p}} + w_2(n_j) \cdot 1 \cdot \dot{\mathbf{p}} \\ &= (w_1(n_j) + w_2(n_j))\dot{\mathbf{p}} = 1 \cdot \dot{\mathbf{p}} = \dot{\mathbf{p}}. \end{aligned} \quad (7)$$

As DN updates at least twice in the unit time, Y area is updated again for the second DN update. But X and Z retain their values within each unit time, per simulation rule. Thus, the Y winner is still the same new neuron and its vector still does not change as the above expression is still true. Thus, properties 1 and 2 are true for the first two DN updates within $(t, t+1]$.

Next consider Z . Z retains its values in the first DN update, per hypothesis. For the 2nd DN update, the response of Z is regarded the DN’s Z output for this unit time, which uses the above Y response as illustrated in Fig. 4. In Case 1, Z must be supervised for this second DN update within the unit time. According to the binary- p mapping from the supervised $q(t+1)$, Eq. (3) is performed for up to p Z neurons:

$$\mathbf{v}_j \leftarrow w_1(n_j)\mathbf{v}_j + w_2(n_j) \cdot 1 \cdot \dot{\mathbf{p}}. \quad (8)$$

Note that Z has only bottom input $\mathbf{p} = \mathbf{y}$ and the normalized vector $\dot{\mathbf{p}}$ is binary. That is, only one component (the new one)

in $\dot{\mathbf{p}}$ is 1 and all other components are zeros. All Z neurons do not link with this new Y neuron before the 2nd DN update. Consider two subcases, subcase (1.a) the Z neuron should fire at the end of this unit time, and subcase (1.b) the Z neuron should not fire.

Subcase (1.a): the Z neuron should fire. All Z neurons that should fire, up to p of them, are supervised to fire for the 2nd DN update by the Z area function. Suppose that a supervised-to-fire Z neuron has a synapse vector $\mathbf{v} = (p_1, p_2, \dots, p_c)$ with the new p_c just initialized to be 0 since the new Y neuron $j = c$ now fires. From the hypothesis, $p_i = f_i/n$, $i = 1, 2, \dots, c-1$. But, according to the Z initialization in GDN, $p_c = 0$ for the new dimension initialization. Then from $0 = p_c = f_c/n$, we have $f_c = 0$ which is correct for f_c . From Eq. (3), the c -th component of \mathbf{v} is

$$v_c \leftarrow \frac{n}{n+1} \cdot \frac{f_c}{n} + \frac{1}{n+1} \cdot 1 \cdot 1 = \frac{f_c + 1}{n+1} = \frac{1}{n+1}. \quad (9)$$

which is the correct count for the new v_c , and the other components of \mathbf{v} are

$$v_i \leftarrow \frac{n}{n+1} \cdot \frac{f_i}{n} + \frac{1}{n+1} \cdot 1 \cdot 0 = \frac{f_i + 0}{n+1} = \frac{f_i}{n+1}, \quad (10)$$

for all $i = 1, 2, \dots, c-1$, which is also the correct count for other components of the \mathbf{v} synaptic vector. Every firing Z neuron advances its age by 1 and correctly counts the firing of the new c -th Y neuron. As Y response does not change for more DN updates within $(t, t+1]$ and the firing Y neuron meets a positive $1/n_j$ weight to the firing Z neuron with

age n_j , the Z area does not need to be supervised after the second DN update within $(t, t + 1]$.

Subcase (1.b): the Z neuron should not fire. All Z neurons that should not fire must be supervised to be zero (not firing). All such Z neurons cannot be linked with the new Y neuron before, as it was not present. The new added weight for this new Y neuron is initialized to 0 in the Z area function. All these non-firing neurons keep their counts and ages unchanged. As Y response does not change for more DN updates within $(t, t + 1]$, the Z area does not need to be supervised after the second DN update within $(t, t + 1]$, since the only firing Y neuron meets a 0 weight to the Z neuron.

The binary conditioning for Z makes sure that all the Z neurons that have a positive pre-response to fire fully. That is, the properties 3 and 4 are true from the first two ND updates within $(t, t + 1]$.

Case 2: old Y input. First consider Y . To Y , $\mathbf{p}(t) = (\mathbf{x}(t), \mathbf{z}(t))$ has been an input before. From the hypothesis, the winner Y neuron j exactly matches $\dot{\mathbf{p}}(t)$, with $\mathbf{v}_j = \dot{\mathbf{p}}(t)$. Eq. (7) still holds using the inductive hypothesis, as the winner Y neuron fires only for a single $\dot{\mathbf{p}}$ vector. Thus, properties 1) and 2) are true from the first ND update within $(t, t + 1]$.

Next consider Z . Z retains its previous vector values in the first DN update, per hypothesis. In the 2nd DN update, the transition is not new, we show that Z does not need to be supervised during the unit time $(t, t + 1]$ to fire perfectly. From Eq. (1), the Z pre-response is computed by

$$r(\mathbf{v}_b, \mathbf{b}) = \frac{\mathbf{v}_b}{\|\mathbf{v}_b\|} \cdot \frac{\mathbf{b}}{\|\mathbf{b}\|} = \frac{\mathbf{v}_b}{\|\mathbf{v}_b\|} \cdot \frac{\mathbf{y}}{\|\mathbf{y}\|} \quad (11)$$

where $\dot{\mathbf{y}}$ is binary with only a single positive component and t is absent as Z does not have a top-down input. Suppose that Y neuron j fired in the first DN update. From the hypothesis, every Z neuron has a synaptic vector $\mathbf{v} = (p_1, p_2, \dots, p_c)$, where $p_j = f_j/n$ counting up to time t , where f_i is the observed frequency (occurrences) of Y neuron j firing, $i = 1, 2, \dots, c$, and n is the total number of times the Z neuron has fired. Consider two sub-cases: (2.a) the Z neuron should fire according to the transition, and (2.b) the Z neuron should not.

For sub-case (2.a) where the Z neuron should fire, we have

$$\begin{aligned} r(\mathbf{v}_b, \mathbf{b}) &= r(\mathbf{v}, \mathbf{y}) = \dot{\mathbf{v}} \cdot \dot{\mathbf{y}} = \frac{p_j}{\|\mathbf{v}\|} \cdot 1 = \frac{p_j}{\|\mathbf{v}\|} \\ &= \frac{f_j/n}{\|\mathbf{v}\|} = \frac{f_j}{n\|\mathbf{v}\|} > 0 \end{aligned}$$

because the Z neuron has been supervised at least the first time for this transition and thus $f_j \geq 1$. We conclude that the Z neuron guarantees to fire at 1 after its binary conditioning. From Eq. (3), the j -th component of \mathbf{v} is:

$$v_j \leftarrow \frac{n}{n+1} \cdot \frac{f_j}{n} + \frac{1}{n+1} \cdot 1 \cdot 1 = \frac{f_j + 1}{n+1} \quad (12)$$

which is the correct count for the j -th component, and the other components of \mathbf{v} are:

$$v_i \leftarrow \frac{n}{n+1} \cdot \frac{f_i}{n} + \frac{1}{n+1} \cdot 1 \cdot 0 = \frac{f_i + 0}{n+1} = \frac{f_i}{n+1}, \quad (13)$$

for all $i \neq j$, which is also the correct count for all other components in \mathbf{v} . The Z neuron does not need to be supervised after the second DN update within $(t, t + 1]$ but still keeps firing. This is what we want to prove for property 3 for every firing Z neuron.

Next consider sub-case (2.b) where the Z neuron should not fire. Similarly we have $r(\mathbf{v}_b, \mathbf{b}) = r(\mathbf{v}, \dot{\mathbf{y}}) = f_j/(n\|\mathbf{v}\|) = 0$, from the hypothesis that this Z neuron fires correctly up to time t and thus we must have $f_j = 0$. Thus, they do not fire, change their weights, or advance their ages. The Z neuron does not need to be supervised after the second DN update within $(t, t + 1]$ but keeps not firing. This is exactly what we want to prove for property 3 for every non-firing Z neuron.

Combining the sub-cases (2.a) and (2.b), all the Z neurons act perfectly and the properties 3 and 4 are true for the first two DN updates. We have proved for Case 2, old Y input.

Therefore, the properties 1, 2, 3, 4 are true for first two DN updates. If DN has time to continue to update before time $t + 1$, we see that we have always Case 2 for Y and Z within the unit time and Y and Z retain their responses since the input \mathbf{x} retains its vector value. Thus, the properties 1, 2, 3, 4 are true for all DN updates within $(t, t + 1]$.

According to the principle of induction, we have proved that the properties 1, 2, 3 and 4 are all true for all t . ■

D. Theorem 1: DN simulates FA incrementally, immediately, and error-free

Using the above lemma, we are ready to prove:

Theorem 1 (Simulate any FA as scaffolding): The general-purpose DP incrementally grows a GDN to simulate any given FA $M = (Q, \Sigma, q_0, \delta, A)$, error-free and on the fly, if the Z area of the DN is supervised when the DN observes each new state transition from the FA. The learning for each state transition completes within two network updates. There is no need for a second supervision for the same state transition to reach error-free future performance. The number of Y neurons in the DN is the number of state transitions in the FA.

Proof: Run the given FA and the GDN at discrete time t , $t = 1, 2, \dots$. Using the lemma above, each state transition $q \xrightarrow{\sigma} q'$ is observed by the DN via the mappings m_x and m_z . Update the DN at least twice in each unit time. In DN, if $\mathbf{p} = (\mathbf{z}, \mathbf{x})$ is a new vector to Y , Y adds a new neuron. Further, from the proof of the above lemma, we can see that as soon as each transition in FA has been taught, the DN has only Case 2 for the same transition in the future, which means that no need for second supervision for any transition. Also from the proof of the lemma, the number of Y neurons corresponds to the number of state transitions in the FA. ■

If the training data set is finite and consistent (the same (q, σ) must go to the unique next state q'), re-substitution test (using the training set) corresponds to simulating an FA using pattern codes. Theorem 1 states that for the DGN any re-substitution test for consistent training data is always immediate and error-free. Conventionally, this will mean that the system over-fits data as its generalization will be poor.

However, the DGN does not over-fit data as the following Theorem 2 states, since the nature of its parameters is optimal and the size of the parameter set is dynamic. In other words, it is optimal for disjoint tests.

Definition 6 (Grounded DN): Suppose that the symbol-to-vector mapping for the DN is consistent with the real sensor of the a real-world agent (robot or animal), namely, each symbol σ for FA is mapped to an sub-image \mathbf{x} from the real sensor, excluding the parts of the irrelevant background in the scene. Then the DN that has been trained for the FA is called grounded.

For a grounded DN, the SN is a human knowledge abstraction of the real world. After training, a grounded DN can run in the real physical world, at least in principle. However, as we discussed above, the complexity of symbolic representation for Σ and Q is exponential in the number of concepts. Therefore, it is intractable for any SN to sufficiently sample the real world since the number of symbols required is too many for a realistic problem. The fact that there are enough symbols to model the real world causes the symbolic system to be brittle. All the probability variants of FA can only adjust the boundaries between any two nearby symbols, but the added probability cannot resolve the fundamental problem of the lack of sufficient number of symbols.

E. Theorem 2: DN generalizes optimally while frozen

The next theorem states how the frozen GDN generalizes for infinitely many sensory inputs.

Theorem 2 (DN generalization while frozen): Suppose that after having experienced all the transitions of the FA, from time $t = t_0$ the GDN turns into a DN that

- 1) freezes: It does not generate new Y neurons and does not update its adaptive part.
- 2) generalizes: It continues to generate responses by taking sensory inputs not restricted to the finite ones for the FA.

Then the DN generates the Maximum Likelihood (ML) action $\mathbf{z}_n(t)$, recursively, for all integer $t > t_0$:

$$n(t) = \arg \max_{\mathbf{z}_i \in Z} h(\dot{\mathbf{p}}(t-1)|\mathbf{z}_i(t), \mathbf{z}(t-1)). \quad (14)$$

where the probability density $h(\dot{\mathbf{p}}(t-1)|\mathbf{z}_i(t), \mathbf{z}(t-1))$ is the probability density of the new last observation $\dot{\mathbf{p}}(t-1)$, with the parameter vector \mathbf{z}_i , conditioned on the last executed action $\mathbf{z}(t-1)$, based on its experience gained from learning the FA.

Proof: Reuse the proof of the lemma. Case 1 does not apply since the DN does not generate new neurons. Only Case 2 applies.

First consider Y . Define c Voronoi regions in $X \times Z$ based on now frozen $V = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_c)$, where each R_j consisting of $\dot{\mathbf{p}}$ vectors that are closer to $\dot{\mathbf{v}}_j$ than to other $\dot{\mathbf{v}}_i$:

$$R_j = \{\dot{\mathbf{p}} \mid j = \arg \max_{1 \leq i \leq c} \dot{\mathbf{v}}_i \cdot \dot{\mathbf{p}}\}, j = 1, 2, \dots, c.$$

Given observation $\mathbf{p}(t-1)$, V has two sets of parameters, the X synaptic vectors and the Z synaptic vectors. They are frozen.

According to the dependence of parameters in DN, first consider consider c events for area Y : $\dot{\mathbf{p}}(t-1)$ falls into R_i , $i = 1, 2, \dots, c$ partitioned by the c Y vectors in V . The conditional probability density $g(\dot{\mathbf{p}}(t-1)|\dot{\mathbf{v}}_i, \mathbf{z}(t-1))$ is zero if $\dot{\mathbf{p}}(t-1)$ falls out of the Voronoi region of $\dot{\mathbf{v}}_i$:

$$g(\dot{\mathbf{p}}(t-1)|\mathbf{v}_i, \mathbf{z}(t-1)) = \begin{cases} g_i(\dot{\mathbf{p}}(t-1)|\mathbf{v}_i, \mathbf{z}(t-1)) & \text{if } \dot{\mathbf{p}}(t-1) \in R_i \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

where $g_i(\dot{\mathbf{p}}(t-1)|\mathbf{v}_i, \mathbf{z}(t-1))$ is the probability density within R_i . Note that the distribution of $g_i(\dot{\mathbf{p}}(t-1)|\mathbf{v}_i, \mathbf{z}(t-1))$ within R_i is irrelevant as long as it integrates to 1.

Note that $\mathbf{p}(t-1) = (\mathbf{x}(t-1), \mathbf{z}(t-1))$. Given $\dot{\mathbf{p}}(t-1)$, the ML estimator for the binary vector $\mathbf{y}_j \in E_1^c$ needs to maximize $g(\dot{\mathbf{p}}(t-1)|\mathbf{v}_i, \mathbf{z}(t-1))$, which is equivalent to finding

$$j = \arg \max_{1 \leq i \leq c} g(\dot{\mathbf{p}}(t-1)|\mathbf{v}_i, \mathbf{z}(t-1)) = \arg \max_{1 \leq i \leq c} \dot{\mathbf{v}}_i \cdot \dot{\mathbf{p}}(t-1), \quad (16)$$

since finding the ML estimator j for Eq. (15) is equivalent to finding the Voronoi region to which $\dot{\mathbf{p}}(t-1)$ belongs to. This is exactly what the Y area does, supposing $k = 1$ for top- k competition.

Next, consider Z . The set of all possible binary-1 Y vectors and the set of producible binary- p Z vectors have a one-to-one correspondence: \mathbf{y}_j corresponds to \mathbf{z}_n if and only if the single firing neuron in \mathbf{y}_j has non-zero connections to all the firing neurons in the binary- p \mathbf{z}_n but not to the non-firing neurons in \mathbf{z}_n . Namely, given the winner Y neuron j , the corresponding $\mathbf{z} \in Z$ vector is deterministic. Furthermore, for each Y neuron, there is only unique \mathbf{z} because of the definition of FA. Based on the definition of probability density, we have:

$$g(\dot{\mathbf{p}}(t-1)|\mathbf{v}_j, \mathbf{z}(t-1)) = h(\dot{\mathbf{p}}(t-1)|\mathbf{z}_n(t), \mathbf{z}(t-1))$$

for every \mathbf{v}_j corresponding to $\mathbf{z}_n(t)$. Thus, when the DN generates $\mathbf{y}(t-0.5)$ in (16) for ML estimate, its Z area generates ML estimate $\mathbf{z}_n(t)$ that maximizes (14). ■

F. Theorem 3: DN thinks optimally

There seems no more proper terms to describe the nature of the DN operation other than “think.” The thinking process by the current basic version of DN seems similar to, but not exactly the same as, that of the brain. At least, the richness of the mechanisms in DN that has demonstrated experimentally to be close to that of the brain.

Theorem 3 (DN generalization while updating): Suppose that after having experienced all the transitions of the FA, from time $t = t_0$ the GDN turns into a DN that

- 1) fixes its size: It does not generate new Y neurons
- 2) adapts: It updates its adaptive part $N = (V, A)$.
- 3) generalizes: It continues to generate responses by taking sensory inputs not restricted to the finite ones for the FA.

Then the DN “thinks” (i.e., learns and generalizes) recursively and optimally: For all integer $t > t_0$, the DN recursively generates the Maximum Likelihood (ML) response $\mathbf{y}_j(t-0.5) \in E_1^c$: with

$$j = \arg \max_{1 \leq i \leq c} g(\hat{\mathbf{p}}(t-1) | \hat{\mathbf{v}}_i(t-1), \mathbf{z}(t-1)) \quad (17)$$

where $g(\hat{\mathbf{p}}(t-1) | \hat{\mathbf{v}}_i(t-1), \mathbf{z}(t-1))$ is the probability density, conditioned on $\hat{\mathbf{v}}_i(t-1), \mathbf{z}(t-1)$. And the Z has the pre-response vector $\mathbf{z}(t) = (r_1, r_2, \dots, r_{c(Z)})$, where $r_n, n = 1, 2, \dots, c(Z)$, is the conditional probability for the n -th Z neuron to fire:

$$r_n = p_{nj}(t) = \text{Prob}(j\text{-th } Y \text{ neuron fires at time } t-0.5 \mid n\text{-th } Z \text{ neuron fires at time } t). \quad (18)$$

The firing of each Z neuron has a freedom to choose a binary conditioning method to map the above the pre-response vector $\mathbf{z} \in R^{c(Z)}$ to the corresponding binary vector $\mathbf{z} \in B^{c(Z)}$.

Proof: Again, reuse the proof of the lemma with the synaptic vectors of Y to be $V(t-1) = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_c)$ now adapting.

First consider Y . Eq (16) is still true as this is what DN does but V is now adapting. The probability density in Eq. (15) is the currently estimated version based on past experience but V is now adapting. Then, when $k = 1$ for top- k Y area competition, the Y response vector $\mathbf{y}_j(t-0.5) \in E_1^c$ with j determined by Eq. (16) gives Eq.(17). In other words, the response vector from Y area is again the Maximum Likelihood (ML) estimate from the incrementally estimated probability density. The major difference between Eq.(16) and Eq.(17) is that in the latter, the adaptive part of the DN updates.

Next, consider Z . From the proof of the Lemma 1, the synaptic weight between the j -th Y neuron and the n -th Z neuron is

$$p_{nj} = \text{Prob}(j\text{-th } Y \text{ neuron fires in the last DN update} \mid n\text{-th } Z \text{ neuron fires in the next DN update}). \quad (19)$$

The total pre-response for the n -th neuron is

$$r_n = r(\mathbf{v}_n, \mathbf{y}) = \hat{\mathbf{v}}_n \cdot \hat{\mathbf{y}} = p_{nj} y_j = p_{nj} 1 = p_{nj},$$

since the j -th neuron is the only firing Y neuron at this time. The above two expressions give Eq. (18). ■

The last sentence in the theorem gives the freedom for Z to choose a binary conditioning method but a binary conditioning method is required in order to determine which Z neurons fire and all other Z neurons do not. In the brain, neural modulation (e.g., expected punishment, reward, or novelty) discourages or encourages the recalled components of \mathbf{z} to fire.

The adaptive mode after learning the FA is autonomous inside the DN. A major novelty of this theory of thinking is that the structure inside the DN is fully emergent, regulated by the DP (i.e., nature) and indirectly shaped (i.e., nurture) by the external environment.

The neuronal resource of Y gradually re-distribute according to the new observations in $Y \times X$. It adds new context-sensory experience and gradually weights down prior experience. Over the entire life span, more often observed experience and less often observed experience are proportionally represented as the synaptic weights.

However, an adaptive DN does not simply repeat the function of the FA it has learned. Its new thinking experience includes those that are not applicable to the FA. The following cases are all allowed in principle:

(1) Thinking with a “closed eye”: A closed eye sets $\mathbf{x} = \mathbf{u}$ where \mathbf{u} has 0.5 for all its components (all gray image). The DN runs where Y responses mainly to \mathbf{z} as \mathbf{x} has little “preference” in matching.

(2) Thinking with an “open eye”: In the sensory input \mathbf{x} is different from any prior input.

(3) Inconsistent experience: From the same $(\mathbf{z}, \mathbf{x}) \equiv (g, \sigma)$, the next $\mathbf{z}' \equiv g'$ may be different at different times. FA does not allow any such inconsistency. However, the inconsistencies allow occasional mistakes, update of knowledge structures, and possible discovery of new knowledge.

The neuronal resources of Y gradually re-distribute according to the new context-motor experience in $Y \times Z$. The learning rate $w_2(n_j) = 1/n_j$ amounts to equally weighted average for past experience by each neuron. Weng & Luciw 2009 [7] investigated amnesic average to give more weight to recent experience.

In the developmental process of a DN, there is no need for a rigid switch between FA and the real-world learning.

The binary conditioning is suited only when Z is supervised according to the FA to be simulated. As the “thinking” of the DN is not necessarily correct, it is not desirable to use the binary conditioning for Z neurons.

The thinking process by the current basic version of DN seems similar to, but not exactly the same as, that of the brain. At least, the richness of the mechanisms in DN is not yet close to that of the brain. For example, the DN here does not use neuromodulators so it does not prefer any signals from receptors (e.g., sweet vs. bitter).

In conclusion, the above analysis and proofs have established the three theorems.

REFERENCES

- [1] R. Desimone and J. Duncan. Neural mechanisms of selective visual attention. *Annual Review of Neuroscience*, 18:193–222, 1995.
- [2] G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006.
- [3] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Boston, MA, 2006.
- [4] M. Minsky. Logical versus analogical or symbolic versus connectionist or neat versus scruffy. *AI Magazine*, 12(2):34–51, 1991.
- [5] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, June 13, 1996.
- [6] J. Weng. Three theorems: Brain-like networks logically reason and optimally generalize. In *Proc. Int’l Joint Conference on Neural Networks*, pages 2983–2990, San Jose, CA, July 31 - August 5, 2011.
- [7] J. Weng and M. Luciw. Dually optimal neuronal layers: Lobe component analysis. *IEEE Trans. Autonomous Mental Development*, 1(1):68–85, 2009.

Neural Modulation for Reinforcement Learning in Developmental Networks Facing an Exponential No. of States

Hao Ye*, Xuanjing Huang*, and Juyang Weng*[†]

*School of Computer Science, Fudan University, Shanghai, 201203 China

[†]Department of Computer Science and Engineering, Cognitive Science Program, Neuroscience Program, Michigan State University, East Lansing, MI, 48824 USA

Email: 10110240032@fudan.edu.cn, xjhuang@fudan.edu.cn, weng@cse.msu.edu

Abstract—Suppose that a developmental agent (animal or machine) has c concepts to learn and each concept has v possible values. The number of states is then v^c , exponential in the number of possible concepts. This computational complexity is well known to be intractable. In artificial intelligence (AI), human handcrafting of symbolic states has been adopted to reduce the number of states, relying on human intuition about the required states of a given task. This paradigm has resulted in the well-known high brittleness because of the inability of the human designer to check the validity of his state reduction for the system to correctly go through an exponential number of paths of state transitions (e.g., in graphic models). In this reported work, we study how a Developmental Network (DN) as an emergent and probabilistic finite automaton (FA) that enables its states to emerge automatically — only those that are experienced in its “life” — greatly reducing the number of actual states. In order to avoid the requirement for the human teacher to specify every state in online teaching (i.e., action in DN), we allow the human teacher to give scores to evaluate the displayed actions (i.e., reinforcement learning), modeling the serotonin system for punishments and the dopamine system for rewards. Due to the need of ground truth for performance evaluation which is hard to come by in the real world, we used a simulation environment described as a game setting, but the methodology is applicable to a real-world developmental robot and also our computational understanding how an animal develops its skills.

I. INTRODUCTION

THE field of natural and artificial intelligence has two schools: symbolic and emergent [1]. The symbolic school has the advantage of intuitiveness in design, but faces the well recognized high brittleness due to the exponential complexity. The emergent school (also known as neural networks) has the advantage of “analogy” but faces the lack of a sufficient capability of abstraction. The recent brain-inspired Developmental Network (DN) theory [2] has bridged the gap between the two schools — the DN learns any complex FA through the observation of the input-output pairs while the FA operates. The FA can be considered the human ontology (common sense knowledge). The learning of the

Hao Ye and Juyang Weng are with School of Computer Science, Fudan University, Shanghai, 200433, China, (email: 10110240032@fudan.edu.cn, weng@cse.msu.edu) and Juyang Weng is also with Department of Computer Science and Engineering, Michigan State University, East Lansing, Michigan, 48824, USA; The authors are grateful to Yin Liu for her participation in the design of game scripts, sensory port, the effector port, art resource and to Hao Dong for his management and support for the game group in the GenericNetwork Inc. The authors would like to thank Bilei Zhu at Fudan University for his help to draw the diagrams.

FA by the DN is incremental, immediate, free of error for not only the input-output sequences that it has observed but also exponentially many input-output sequences that it has not observed but state-equivalent [2]. Furthermore, the DN is optimal in the sense of maximum likelihood when it faces an infinite number of sensorimotor experiences of the real physical world. However, the FA is an abstract model, not the real physical or simulated world. Weng et al. 2013 [3] studied DN reinforcement learning for face recognition and a three-agent wandering setting, but the issue of exponential number of possible states was not addressed in those two settings since they only involved one concept (type or heading).

A complex (simulated) environment of EpigenBuddy [4] (see Fig. 1) shows the demand of the exponential number of states. The player should teach the NPC various skills to accomplish various tasks in each game scene. For example, in motor port of our game, we have ten concept zones, each has eight neurons on average, then we totally have 8^{10} conception combinations which are more than one billion, beyond most game engines’ processing capability. Five concept zones of motor port are response for the action behavior: scene, NPC’s orientation, eyes, upper arms and legs. The rest five concept zones are response for art control. TABLE 1 gives those five concepts as five columns and the rows give a few examples from an intractable number of possible states.

TABLE I
STATES OF EPIGENBUDDY

State	Scene	Orientation	Eyes	Upper arms	Legs
S11	Home	Facing closet	Searching	Fumbling	Walking
S12	Home	Front	Exciting	Grab Map	Stand
S21	Outdoor	Facing hill	Searching	Fumbling	Walking
S22	Outdoor	Front	Exciting	Grab the stick	Stand
S31	Tunnel	Facing devil	Searching	Grab the sword	Walking
S32	Tunnel	Facing devil	Track devil	Fighting	Running
S33	Tunnel	Facing devil	Shun devil	Hold sword	Stand

Take the state transition $S_{11} \xrightarrow{\sigma_2} S_{12}$ in scene *home* for instance. In order to present S_{11} which means the NPC is *looking for the map*, its concept combination is: in scene *home*, its orientation concept zone emerges *facing closet* concept, eyes concept zone emerges *searching behavior*, upper arms concept zone presents *fumbling behavior* and legs concept zone emerges *walking straight forward*. The combination of these motor concept zones will trigger the

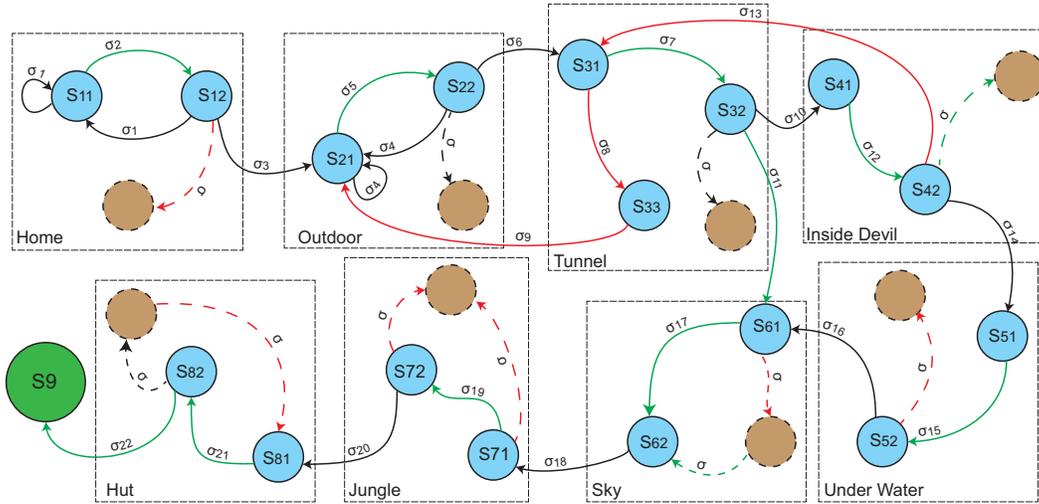


Fig. 1. A simplified emergent FA of the EpigenBuddy game. The blue states are the main states of the game. The brown states encircled by dash line indicate those possible states and transitions without experience and do not emerge in internal representation. The green one is the final state.

game’s cartoon engine to animate the NPC’s *searching map* action. When the NPC receives the input σ_2 , which means the map was detected, the NPC will transit to state S_{12} . The main concept zones responsible for the input are *scene*, *object’s location*, *object’s type*, *pain receptor* and *sweet receptor*. The input σ_2 can be encoded by the concepts of *home*, *the map’s location*, *the maze map*, *no pain*, and *happy feeling*. TABLE II gives five inputs in our game as example.

TABLE II
INPUTS OF NPC

Input	Scene	Object’s Location	Object’s Type	Pain Receptor	Sweet Receptor
σ_1	Home	(0,0)	Nothing	No	No
σ_2	Home	(Location)	Map	No	Yes
σ_3	Home	(Location)	Map	No	No
σ_4	Outdoor	(0,0)	Nothing	No	No
σ_5	Outdoor	(Location)	Stick	No	Yes

Training NPC only by supervised learning is inefficient and boring. By contrast, reinforcement learning, which is more convenient and efficient, is another common way of learning in human society. Reinforcement learning enables human beings to learn autonomously through trial-and-error interactions in the dynamic environment. Figure 2 shows the difference between these two sorts of learning in our game. Supervised learning needs to choose several concepts manually, step by step, even you may have done such supervisions many times. However, reinforcement learning just needs to evaluate the actions of the NPC, such as rewarding by the dropping candy boxes from the sky. Using dropping boxes is to add fun to the game, although it can be rewarded or punished directly.

Two strategies were proposed to implement reinforcement learning. One is to thoroughly search the behavior space to find the optimal behaviors; the other is to use the statistical and dynamic programming methods to evaluate the consequence of all behaviors. But both two kinds of



Fig. 2. Subfigure (a) is supervised learning. The player should choose the concepts listed on the right side manually. Subfigure (b) is reinforcement learning. The player teaches the NPC just through reward boxes dropped from the sky. A reward or punishment that does hit the NPC is sensed by the NPC. The user needs to control the NPC to catch the reward or to avoid the punishment.

methods are symbolic models, they are vulnerable to the exponentially grow of inner states. Reinforcement Learning for neuromorphic networks that is capable of implementing FA is a recent development [5] [2] which shows its capability of dealing with such problem.

Neuromorphic network is a kind of DN that uses the neuron-like unit. It simulates the brain’s neural modulation system, which uses neurotransmitters secreted from the brain to determine human being’s sensations such as like or dislike, happy or depress. Plenty of biological and psychological researches proved that human being has motivation system [6] [7] controlled by neural modulation system. It is exactly based on the motivation system that human beings possess their capability of reinforcement learning. Therefore, an augmented DN with modulation system named neuromorphic DN was proposed.

In this paper, we performed an experiment in our digital game platform to test the effectiveness of our network. We constructed several NPCs based on neuromorphic DN with different parameters and investigated their learning capability and efficiency under different parameters. Learning from

the experimental results, we came to conclusion that the neuromorphic network highly improved the capability of NPCs autonomous learning. The NPC made its optimal decision based on its past experience.

The remainder of this paper is structured as follows. We briefly introduce some related works in section II and DN in section III. In section IV we introduce the updated version of DN with modulation system and its algorithm. We show the experimental results in section V to verify the property of our model.

II. RELATED WORKS

Motivation systems are divided into two genres, symbolic and neuromorphic. Symbolic systems are designed handcrafted, where the meaning of the states is predefined. Neuromorphic means the model uses neuron-like units and the meaning of the states is emergent.

A. Symbolic value systems

Sutton & Barto 1981 [8] modeled rewards as positive values that the system learns to predict. Ogmens work [9] was based on Adaptive Resonance Theory (ART), which took into account not only punishments and rewards, but also the novelty in expected punishments and rewards, where punishments, rewards and novelty are all based on a single value. Almassy et al. 1998 [10], further refined in Sporns et al. 1999 [11], modeled a robotic system where punishments and rewards after interactive trials affect the later behaviors of the robot. Each area in their system is a network, but the features in each area are handcrafted, belonging to our definition of symbolic representations. Kakade & Dayan [12] proposed a dopamine model, which uses novelty and shaping to drive exploration in reinforcement learning, although they did not provide source of information for novelty nor a computational model to measure the novelty. Oudeyer et al. 2007 [13] proposed that the objective functions for a robot uses as a criterion to choose an action fall into three categories, (1) error maximization, (2) progress maximization, and (3) similarity-based progress maximization. Huang & Weng 2007 [14] proposed an intrinsic motivation system that prioritizes three types of information with decreasing urgency: (1) punishment, (2) reward and (3) novelty. As punishment and rewards are typically sparse in time, novelty can provide temporally dense motivation even during early life. Krichmar 2008 [15] provided a survey that includes four types of neural transmitters. Singh et al, 2010 [16] adopted an evolutionary perspective and define a new reward framework that captures evolutionary success across environments.

B. Neuromorphic value systems

Cox & Krichmar 2007 [17] proposed and experimented with an architecture that integrates three types of neurotransmitters, 5-HT, DA and Ach (Acetylcholine) with Ach for increased attention efforts. In their system, the sensory system is emergent, but the behavior system is not. The behavior system has three types of handcrafted behavior states, random exploration, find, and flee. Krichmar 2008

[15] provided a survey which includes four types of neural transmitters: 5HT, DA, Ach, and NE (Norepinephrine).

Weng et al. [18] [5] [3] updated the DN with modulation system and used the new model in navigation and face classification. In navigation, attractor and repulsor are placed in the map to intrigue the releases of dopamine and serotonin. In face classification, there are no supervisions but only answers to the classification results. All the states in the model are fully emergent.

III. DEVELOPMENTAL NETWORK

DN can abstract as well as symbolic networks such as Bayesian Network [19], Markov Decision Process [20]. Because the DN takes not only the sensory port but also motor port as the input.

A. Architecture

A basic DN has three areas, the sensory area X , the internal (“brain”) area Y and the motor area Z . An example of DN is shown in Fig. 3. The internal neurons in Y have bi-directional connection with both X and Z .

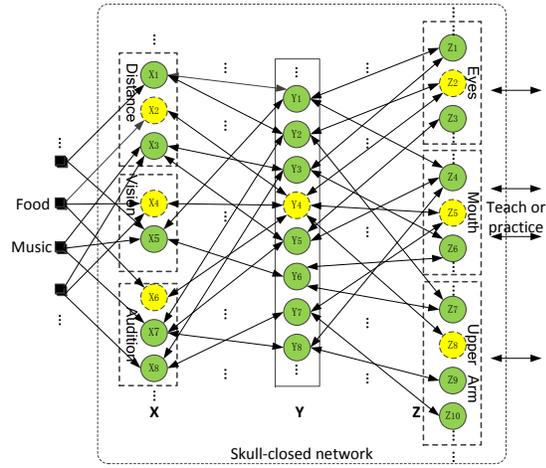
The DP (Developmental Program) for DNs is not task-specific as suggested for the “brain” in [21] (e.g., not concept-specific or problem-specific). In contrast to a static FA, the motor area Z of a DN can be directly observed by the environment (e.g., by the teacher) and thus can be calibrated through interactive teaching from the environment. The environmental concepts are learned incrementally through interactions with the environments. For example, in Fig. 3, the “Food” object makes the pixels 2, 4 and 6 activated and all other green pixels remain normal. However, such an image from the “Food” object is not known during the programming time for the DP.

In principle, the X area can model any sensory modality (e.g., vision, audition, and touch). The motor area Z serves both input and output. When the environment supervises Z , Z is the input to the network. Otherwise, Z gives an output vector to drive effectors (muscles) which act on the real world. The order of areas from low to high is: X, Y , and Z . For example, X provides bottom-up input to Y , but Z gives top-down input to Y .

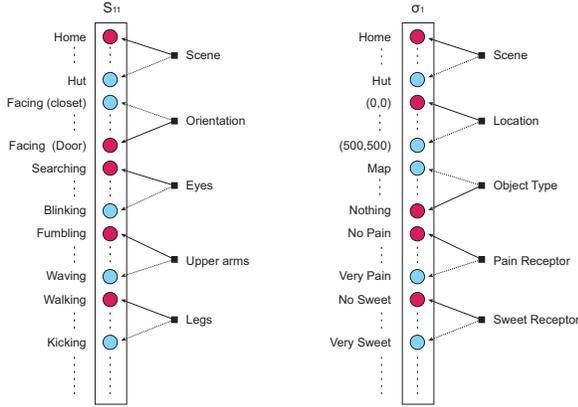
B. DN algorithm

The DN algorithm is as follows. Input areas: X and Z . Output areas: X and Z . The dimension and representation of X and Z areas are hand designed based on the sensors and effectors of the robotic agent or biologically regulated by the genome. Y is skull-closed inside the “brain”, not directly accessible by the external world after the birth.

- 1) At time $t = 0$, for each area A in $\{X, Y, Z\}$, initialize its adaptive part $N = (V, G)$ and the response vector r , where V contains all the synaptic weight vectors and G stores all the neuronal ages. For example, use the generative DN method discussed below.
- 2) At time $t = 1, 2, \dots$, for each A in $\{X, Y, Z\}$ repeat:



(a) State transition of the EpigenBuddy.



(b) Template

Fig. 3. (a) A DN has three parts: X zone response for the sensor, Z zone response for motor, and Y zone correspond to the inner “brain” which has bi-directional connections with both X and Z . In our game, X does not use Y inputs. Pixel in yellow and circled by dash-line means it is activated. Pixels grouped in dotted box are in the same concept zone. (b) An example of a state and an input represented in Z motor and X motor. Pixels in red indicate those concepts are activated.

- a) Every area A performs mitosis-equivalent if it is needed, using its bottom-up and top-down inputs \mathbf{b} and \mathbf{t} , respectively.
- b) Every area A computes its area function f , described below,

$$(\mathbf{r}', N') = f(\mathbf{b}, \mathbf{t}, N)$$

where \mathbf{r}' is its response vector.

- c) For every area A in $\{X, Y, Z\}$, A replaces: $N \leftarrow N'$ and $\mathbf{r} \leftarrow \mathbf{r}'$.

C. Unified DN area function

It is desirable that each “brain” area uses the same area function f , which can develop area specific representation and generate area specific responses. Each area A has a weight vector $\mathbf{v} = (\mathbf{v}_b, \mathbf{v}_t)$. Its pre-response value is:

$$r(\mathbf{v}_b, \mathbf{b}, \mathbf{v}_t, \mathbf{t}) = \hat{\mathbf{v}} \cdot \hat{\mathbf{p}} \quad (1)$$

where $\hat{\mathbf{v}}$ is the unit vector of the normalized synaptic vector $\mathbf{v} = (\hat{\mathbf{v}}_b, \hat{\mathbf{v}}_t)$, and $\hat{\mathbf{p}}$ is the unit vector of the normalized input vector $\mathbf{p} = (\hat{\mathbf{b}}, \hat{\mathbf{t}})$. The inner product measures the degree of match between these two directions $\hat{\mathbf{v}}$ and $\hat{\mathbf{p}}$, because $r(\mathbf{v}_b, \mathbf{b}, \mathbf{v}_t, \mathbf{t}) = \cos(\theta)$ where θ is the angle between two unit vectors $\hat{\mathbf{v}}$ and $\hat{\mathbf{p}}$. This enables a match between two vectors of different magnitudes (e.g., a weight vector from an object viewed indoor to match the same object when it is viewed outdoor). The pre-action value ranges in $[-1, 1]$.

To simulate lateral inhibitions (winner-take-all) within each area A , top k winners fire. Considering $k = 1$, the winner neuron j is identified by:

$$j = \arg \max_{1 \leq i \leq c} r(\mathbf{v}_{bi}, \mathbf{b}, \mathbf{v}_{ti}, \mathbf{t}). \quad (2)$$

The area dynamically scales top- k winners so that the top- k respond with values in $(0, 1]$. For $k = 1$, only the single winner fires with response value $y_j = 1$ (a pike) and all other neurons in A do not fire. The response value y_j approximates the probability for $\hat{\mathbf{p}}$ to fall into the Voronoi region of its $\hat{\mathbf{v}}_j$ where the “nearness” is $r(\mathbf{v}_b, \mathbf{b}, \mathbf{v}_t, \mathbf{t})$.

D. DN learning: Hebbian

All the connections in a DN are learned incrementally based on Hebbian learning — cofiring of the pre-synaptic activity $\hat{\mathbf{p}}$ and the post-synaptic activity y of the firing neuron. If the pre-synaptic end and the post-synaptic end fire together, the synaptic vector of the neuron has a synapse gain $y\hat{\mathbf{p}}$. Other non-firing neurons do not modify their memory. When a neuron j fires, its firing age is incremented $n_j \leftarrow n_j + 1$ and then its synapse vector is updated by a Hebbian-like mechanism:

$$\mathbf{v}_j \leftarrow w_1(n_j)\mathbf{v}_j + w_2(n_j)y_j\hat{\mathbf{p}} \quad (3)$$

where $w_2(n_j)$ is the learning rate depending on the firing age (counts) n_j of the neuron j and $w_1(n_j)$ is the retention rate with $w_1(n_j) + w_2(n_j) \equiv 1$. The simplest version of $w_2(n_j)$ is $w_2(n_j) = 1/n_j$ which corresponds to:

$$\mathbf{v}_j^{(i)} = \frac{i-1}{i}\mathbf{v}_j^{(i-1)} + \frac{1}{i}\mathbf{p}(t_i), i = 1, 2, \dots, n_j,$$

where t_i is the firing time of the post-synaptic neuron j . The above is the recursive way of computing the batch average:

$$\mathbf{v}_j^{(n_j)} = \frac{1}{n_j} \sum_{i=1}^{n_j} \mathbf{p}(t_i)$$

where is important for the proof of the optimality of DN in [22].

IV. MODULATION SYSTEM

Psychological studies have proved rich evidence about the existences of the motivational system [23] [7], which is biologically closely related to modulatory systems [15]. Without motivation system, the brain is just a mapper from input to output without the sense of what is like or dislike. Lack of this motivation, the brain will lose its ability of learning autonomously.

The functions of neural modulation are implemented by neurotransmitters (e.g., serotonin and dopamine). Instead of carrying out detailed sensorimotor information, neurotransmitters often perform regulatory functions, modulating the postsynaptic neurons (e.g., the cerebral cortex, and the thalamus) so that they become more or less excitable or make fewer or more connections. There are many kinds of neurotransmitters. In our model we only model two of them, which are dopamine and serotonin because their functions are well disclosed.

Dopamine (DA) is associated with reward, which indicates pleasure, wanting, etc. It is released from the substantia nigra and the VTA. If an agent is expecting a reward, dopamine is also released.

Serotonin (5-HT) is responses for pain, stress, threats or punishment. Serotonin leads to behavior inhibition and aversion to punishment. It originates in the raphe nuclei (RN) of the brain stem.

A. Models with DA and 5-HT

To model DA and 5-HT, our model needs two more types of neurons:

Dopaminergic neurons are those neurons that are sensitive to dopamine. Firing of these neurons indicates pleasure.

Serotonergic neurons are those neurons that are sensitive to serotonin. Firing of these neurons indicates stress.

The dopamine and serotonin appear to have effect on neurons in its motor area. When a motor neuron receives dopamine, it will be more likely to fire. While the neuron receives serotonin, it will be less likely to fire. Therefore, when an action gets the reward or avoids danger, the brain will release dopamine, then this action will be encouraged to react in the similar circumstance next time. If the action gets punishment, then the possibility of reacting will become smaller in the same scenario next time.

Link all sweet receptors with hypothalamus (HT) – represented as an area, which has the same number of neurons as the number of sweet receptors. Every neuron in the hypothalamus releases only dopamine.

Link all pain receptor with RN located in the brain stem – represented as an area which has the same number of neurons as the number of pain receptor. Every neuron in RN releases only serotonin. Figure 4 presents the architecture of updated DN with modulation system.

The X, Y, Z areas in updated DN are augmented. Sensory area $X = (X_u, X_p, X_s)$ consists of an unbiased array X_u , a pain array X_p and a sweet array X_s . $Y = (Y_u, Y_p, Y_s, Y_{RN}, Y_{HT})$ connects with X , RN and HT as bottom-up and Z as top-down. The motor area is denoted as $Z = (\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_m)$, where m is the number of muxels. Each \mathbf{z}_i has three neurons $\mathbf{z}_i = (z_{iu}, z_{ip}, z_{is})$, where z_{iu}, z_{ip}, z_{is} are unbiased, pain, and sweet respectively, $i = 1, 2, \dots, m$. z_{ip} and z_{is} are serotonin and dopamine collaterals, associated with z_{iu} , as illustrated by the Z area in Fig. 4. In our game setting, an action consists of a number of Z neurons to combine various concepts, so that we allow several z neurons fire simultaneously.

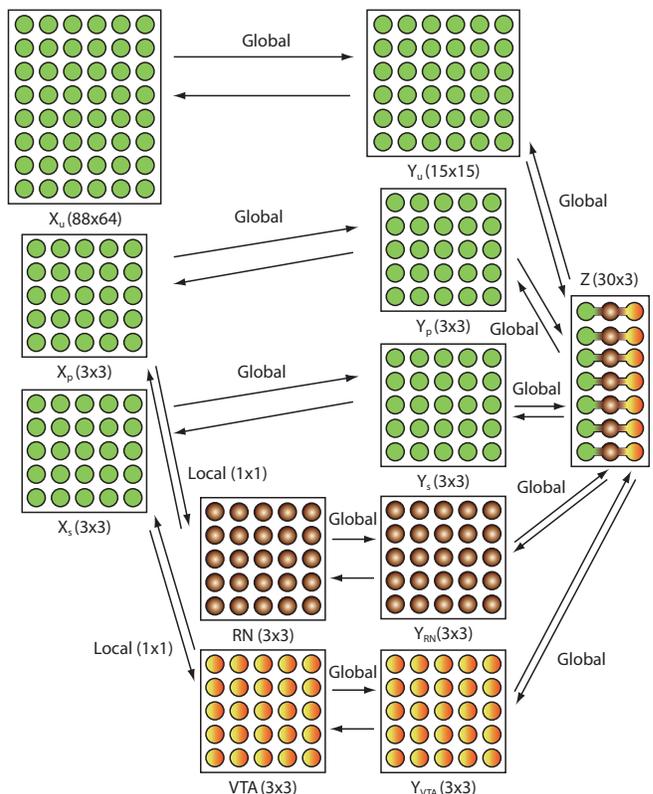


Fig. 4. A DN with 5-HT and DA. The brown color denotes serotonergic neurons. The yellow color denotes dopaminergic neurons. The areas $Y_u, Y_p, Y_s, Y_{RN}, Y_{HT}$ should reside in the same cortical areas, each represented by a different type of neurons, with different neuronal densities.

Whether the action i is released or not depends on not only the response of z_{iu} but also those of z_{ip} and z_{is} . z_{ip} and z_{is} report how much negative value and positive value are associated with the i -th action.

They form a triplet for each action \mathbf{z}_i . We use the following intracellular rule for each motor triplet, assuming that this models the effects of the serotonin and dopamine on the internal mechanisms of a motor neuron.

Definition 1: (collateral Rule): Each motivated action is a vector $\mathbf{z}_i = (z_{iu}, z_{ip}, z_{is})$ in $Z = (\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_m)$, $i = 1, 2, \dots, m$. The response of the action neuron is determined by

$$z_{iu} \leftarrow \max\{z_{iu}(1 + z_{is} - \alpha z_{ip}), 0\} \quad (4)$$

with a large constant $\alpha > 1$, where the parameter α controls the sensitivity to pain. A large α means the agent is more sensitive to pain. Another parameter controlling the sensitivity is the threshold which will be explained in the next section.

B. Operation

As an example of a motivational system, let us discuss how such a motivated network realizes instrumental conditioning, a well known animal model of reinforcement learning discussed in psychology [24].

We only first consider a one-step delay in the following model here. We use an arrow (\rightarrow) to denote causal change over time.

Suppose that an action \mathbf{z}_a leads to pain but another action \mathbf{z}_b leads to sweet. Using our notation, we have

$$\begin{aligned} (\mathbf{x}(t_1); \mathbf{z}(t_1)) &= (\mathbf{x}_u(t_1), \mathbf{o}(t_1), \mathbf{o}(t_1); \mathbf{z}_a(t_1)) \\ &\rightarrow (\mathbf{x}_u(t_1+1), \mathbf{x}_p(t_1+1), \mathbf{o}(t_1+1); \mathbf{z}_{a,p}(t_1+1)) \end{aligned}$$

and

$$\begin{aligned} (\mathbf{x}(t_2), \mathbf{z}(t_2)) &= (\mathbf{x}_u(t_2), \mathbf{o}(t_2), \mathbf{o}(t_2); \mathbf{z}_b(t_2)) \\ &\rightarrow ((\mathbf{x}_u(t_2+1), \mathbf{o}(t_2+1), \mathbf{x}_s(t_2+1); \mathbf{z}_{b,s}(t_2+1)) \end{aligned}$$

where p and s indicate pain and sweet and \mathbf{o} denotes a zero vector. The vectors \mathbf{z}_a and \mathbf{z}_b represent two different actions with different i 's for \mathbf{z}_i .

In our example above, let $\mathbf{z}_a = (1, 0, 0)$ and $\mathbf{a}_b = (1, 0, 0)$, but $\mathbf{z}_{a,p} = (1, 1, 0)$ and $\mathbf{z}_{b,s} = (1, 0, 1)$.

Next, the agent runs into a similar scenario $\mathbf{x}'_u \approx \mathbf{x}_u$. $\mathbf{x}' = (\mathbf{x}'_u, \mathbf{o}, \mathbf{o})$ is matched by the same vector \mathbf{y} as $\mathbf{x} = (\mathbf{x}_u, \mathbf{o}, \mathbf{o})$. Through the \mathbf{y} vector response in Y , the motor area comes up with two actions $\mathbf{z}_{a,p} = (1, 1, 0)$ and $\mathbf{z}_{b,s} = (1, 0, 1)$. Using our collateral rule, \mathbf{z}_a is suppressed and \mathbf{z}_b is executed.

We note that the above discussion only spans one unit time of the network update. However, the network can continue to predict:

$$(\mathbf{x}(t), \mathbf{z}(t)) \rightarrow (\mathbf{x}(t+1), \mathbf{z}(t+1)) \rightarrow (\mathbf{x}(t+2), \mathbf{z}(t+2))$$

and so on. This seems a biologically plausible way of dealing with delayed reward.

C. Action Procedure

A protocol of training and acting is as follows:

- 1) Learning Pain: $(\mathbf{x}_u, \mathbf{o}, \mathbf{o}, \mathbf{z}_a) \rightarrow (\mathbf{x}_u, \mathbf{x}_p, \mathbf{o}, \mathbf{z}_{u,p})$
- 2) Learning sweet: $(\mathbf{x}_u, \mathbf{o}, \mathbf{o}, \mathbf{z}_b) \rightarrow (\mathbf{x}_u, \mathbf{o}, \mathbf{x}_s, \mathbf{z}_{u,s})$
- 3) Pain avoidance and pleasure seeking:

$$(\mathbf{x}_u, \mathbf{o}, \mathbf{o}; \mathbf{o}) \rightarrow (\mathbf{x}_u, \mathbf{o}, \mathbf{o}; \mathbf{z}_{a \cup b}) \rightarrow (\mathbf{x}_u, \mathbf{x}_s, \mathbf{o}; \mathbf{z}_{b,s})$$

where $\mathbf{z}_{a \cup b} \in Z$ denotes a response vector where both \mathbf{z}_a and \mathbf{z}_b are certain but with different collaterals.

V. EXPERIMENT

A. Experiment setting

We use the first scene, which is a 500x500 square map in a hall, as the test platform. A pair of coordinate values defines the position of one agent. The agent can move in 8 directions in the map at different speed controlled by step size. There is an elf as the attractor, and a devil as the repulsor in the map. Elf and devil are always wandering in the map randomly. When these two kinds of agents run into the NPC, they interact with the NPC then give feedbacks to the NPC's current action. The elf is always rewarding the NPC, while the devil is always punishing the NPC.

There are some additional concepts we should use in this game. For the sensory concept zones, we usually use the distance and vigilance concepts to measure the danger. Distance sensors receive the positions of devil and elf, and calculate the distances to these two agents. Vigilance

represents the properties of the detected agents, danger or sweet. Another important parameter of NPC representing the sensitivities to danger and sweet is the distance thresholds to detect danger and sweet (denoted by τ_{devil} and τ_{elf}). These thresholds discriminate the NPC as blunt, sensitive to danger, sensitive to desire, or sensitive to both. TABLE III lists all the threshold settings. For the motor concept zones, the body and leg concept zones are often used, which lead to three useful actions: wandering, chasing and escaping.

TABLE III
SENSITIVITY THRESHOLDS TO DANGER AND SWEET

Threshold	Danger	Sweet
Blunt	50	50
Sensitive to Danger	200	50
Sensitive to Sweet	50	200
Sensitive to Both	200	200

NPC with modulation system also has pain and sweet receptors. Once the NPC crosses the threshold when it is approaching to the devil or elf, its pain or sweet receptor will be triggered and the NPC starts to release serotonin or dopamine. Although the influences of serotonin and dopamine are complicated, we consider that the effect of the serotonin overwhelms the effect of dopamine in our game scenario, which means the NPC will in horror as long as its distance to the devil is less than the threshold. TABLE IV lists all status in different distances to the elf and devil.

TABLE IV
SITUATIONS OF NPC

Distance	$< \tau_{devil}$	$> \tau_{devil}$
$< \tau_{elf}$	In horror	In desire
$> \tau_{elf}$		In leisure

B. Analysis

First of all, we investigate different kinds of NPCs with or without modulation system, and show their different behaviors. The NPCs with modulation system are learning to get close to the friend and shun the foe quickly, which can be learned from the distance distribution in Fig. 5. All four NPCs with modulation system of different level: *blunt*, *sensitive to friend*, *sensitive to enemy*, *sensitive to both* have regular distance distributions, while the NPC without modulation system (*no brain*) has a random distance distribution to the friend and foe. In real game setting, the distance lower than 10 to the devil means the NPC will lose its mission and have to restart the game. This implicates the learning efficiency of the NPC without modulation is very low when the players are not teaching the NPC how to deal with the current situation. This circumstances often occurs in the complex game setting where the players cannot even handle the large amount of states.

More specifically, we can learn from the chart that the distances to the friend are below 100, smaller than the distances to the foe, which are between 100 and 200. This

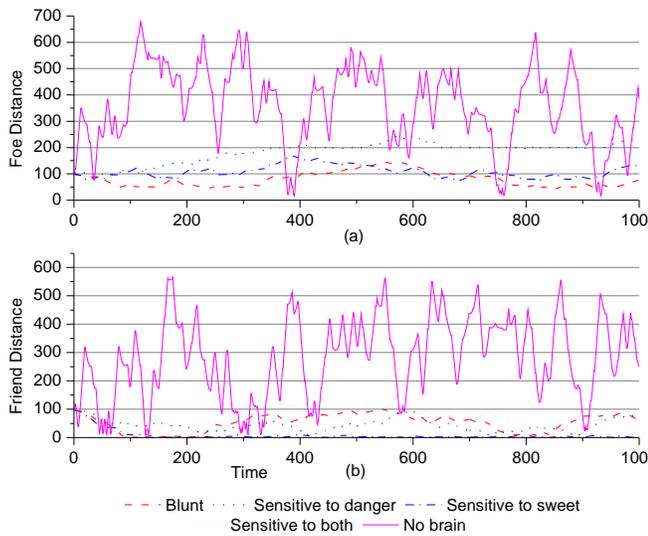


Fig. 5. Distance distribution to friend and foe. Subfigure (a) shows the trends of each kind of NPC’s distance to the foe. Subfigure (b) shows the trends of the distance to friend.

shows the different effects of dopamine and serotonin. When NPC approaches to the elf, it receives the positive feedback, and begins to release dopamine, which encourages the NPC to invoke the action rewarded. While the NPC approaches the devil and crosses the line of threshold, it suffers from punishment, and begins to release serotonin, which invokes the action that decreases the danger.

A detailed study of four different modulation level modulation is illustrated in Fig. 6. The blunt NPC’s thresholds to danger and sweet are both small, this makes the devil can stay close to the NPC. We can see from the upper left sub-figure in Fig. 6, sometimes the distance to the devil stands even around 50. Blunt NPC is also very easy to be got rid of, for example, during the time interval 200 to 800, the elf is manage to escape from the NPC’s chasing.

Shown from the upper right sub-figure of Fig. 6, the danger sensitive NPC has an obvious warning line, which equals to the threshold of danger. Since the NPC is insensitive to the sweet, it will also ignore the elf. We can learn from the sub-figure that the distance to the friend is also fluctuating. This kind of NPC is very vigilant but hard to get reward from the elf.

Opposite to the danger sensitive NPC, the sweet sensitive NPC seldom allows the elf to flee away. The lower left in Fig. 6 reports that the NPC is always catching up the elf and keeping near to it.

The most interesting NPC is the one that sensitive to both danger and sweet. It has the qualities of cautious and spirit of adventure. It will not only keep itself in the safe belt, but also try to give it a shot to chase the elf when in danger. We can find out this phenomenon in lower right sub-figure in Fig. 6 at about time 200.

Figure 7 shows the comparison of average distances of different NPCs. Sensitive to friend NPC has the smallest distance to friend, while the no brain NPC has the largest,

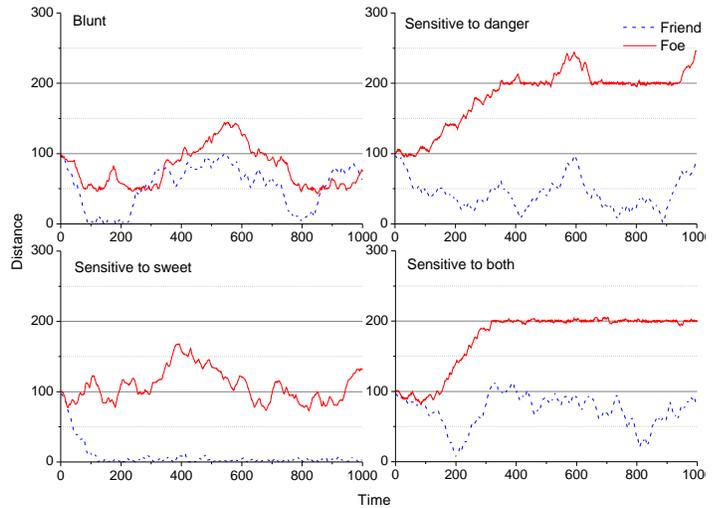


Fig. 6. Distance trends of each kind of NPC. Upper left is the blunt NPC. Upper right is the sensitive to danger NPC. Lower left is the distance distribution of sensitive to sweet NPC. Lower right is the distance distribution of sensitive to both NPC.

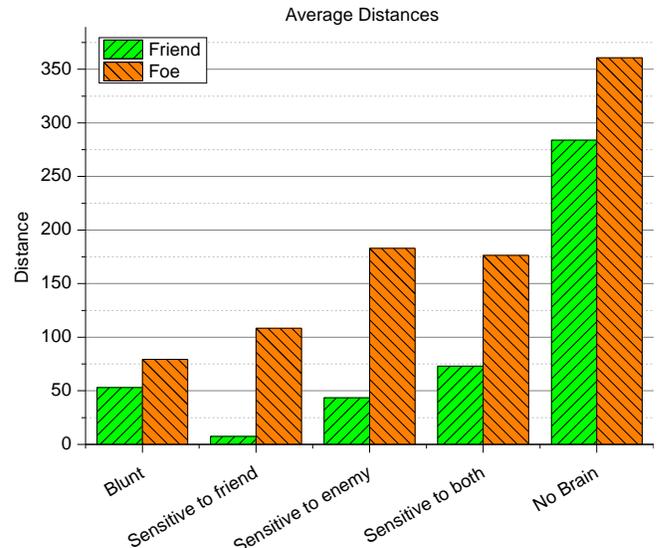
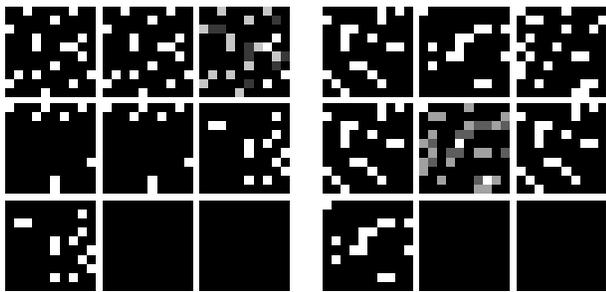


Fig. 7. Average distance of different kinds of NPC

the rest are in middle. It is worth noticing that the average distances to enemy are equal in the case of the NPC, which is sensitive to enemy and which is sensitive to both, because their thresholds of enemy are equal. Their average distances to enemy are both larger than the sensitive to friends. This figure also shows the linear correlation between the threshold and the distance.

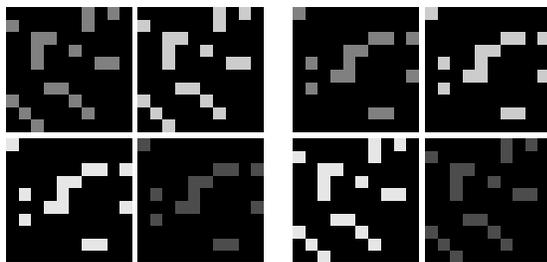
Figure 8 visualizes the weights of neurons in Y area. Each square represents a Y neuron, which has bottom-up and top-down connections to the sensory area and motor area. The Y neuron is activated when the bottom-up and top-down inputs match the weights. Left of Fig. 8 is the bottom-up connection weights of Y, the white spots indicate the weights are equal to 1, black ones are equal to 0, and the gray means between



(a) Bottom-up connection weights (b) Top-down connection weights

Fig. 8. Connection weights of the Y area. Each square represents a Y neuron’s connection weight. Left sub-figure represents the connection weights of bottom-up (from X area). The right sub-figure represents the connection weights of top-down (from Z area).

0 and 1. The right of Fig. 8 is the top-down connections. The last two neurons are not used in our experiment and are all black.



(a) Serotonin Level (b) Dopamine Level

Fig. 9. Neurotransmitter effects on motor area. The left sub-figure shows the weights of Y_{RN} to motor area, represents the effect of serotonin. The right sub-figure shows the weights of Y_{HT} to motor area, represents the effect of dopamine.

Figure 9 reflects the effects of neurotransmitters. The left of the diagram is the effect of the serotonin and the right is the effect of dopamine on motor area. Same patterns but different intensities of the weights can be spotted in the Fig. 9.

VI. CONCLUSIONS

Developmental agents for games seem to open a new avenue for future digital games, with applications in entertainment, education, training, and simulations for agent research. In this paper, we showed that such developmental agents, with fully emergent internal representations (in the network other than X and Z areas) can accommodate a mixture of user-initiated motor-supervised learnings and reinforcement learnings. This new capability for digital games increases the playing value and reduce the human player’s teaching load. However, occasional motor-supervised learnings are still allowed for a quick acquisition of user-desired behaviors. This new type of developmental agents allow sensory-motor skills learned in earlier playing to be automatically applied to later learning and playing. The future work is to fully implement the game and test it in commercial settings.

REFERENCES

- [1] J. Weng, “Symbolic models and emergent models: A review,” *IEEE Trans. Autonomous Mental Development*, vol. 4, no. 1, pp. 29–53, 2012.
- [2] —, “Why have we passed “neural networks do not abstract well”?” *Natural Intelligence: the INNS Magazine*, vol. 1, no. 1, pp. 13–22, 2011.
- [3] J. Weng, S. Paslaski, J. Daly, C. VanDam, and J. Brown, “Modulation for emergent networks: Serotonin and dopamine,” *Neural Networks*, vol. 41, pp. 225–239, 2013.
- [4] H. Ye, X. Huang, and J. Weng, “Inconsistent training for developmental networks and the applications in game agents,” in *Proc. International Conference on Brain-Mind*, East Lansing, Michigan, July 14–15, 2012, pp. 43–59.
- [5] S. Paslaski, C. VanDam, and J. Weng, “Modeling dopamine and serotonin systems in a visual recognition network,” in *Proc. Int’l Joint Conference on Neural Networks*, San Jose, CA, July 31 - August 5, 2011, pp. 3016–3023.
- [6] A. H. Maslow, *Motivation and Personality*, 1st ed. New York: Harper and Row, 1954.
- [7] T. W. Robbins and B. J. Everitt, “Neurobehavioural mechanisms of reward and motivation,” *Current Opinion in Neurobiology*, vol. 6, no. 2, pp. 228–236, April 1996.
- [8] R. Sutton and A. Barto, “Toward a modern theory of adaptive networks: Expectation and prediction,” *Psychological Review*, vol. 88, pp. 135–170, 1981.
- [9] H. Ogmen, “A developmental perspective to neural models of intelligence and learning,” in *Optimality in Biological and Artificial Networks*, D. Levine and R. Elsberry, Eds. Hillsdale, NJ: Lawrence Erlbaum, 1997, p. 363395.
- [10] N. Almassy, G. M. Edelman, and O. Sporns, “Behavioral constraints in the development of neural properties: A cortical model embedded in a real-world device,” *Cerebral Cortex*, vol. 8, no. 4, pp. 346–361, 1998.
- [11] O. Sporns, N. Almassy, and G. Edelman, “Plasticity in value systems and its role in adaptive behavior,” *Adaptive Behavior*, vol. 7, no. 3, 1999.
- [12] S. Kakade and P. Dayan, “Dopamine: generalization and bonuses,” *Neural Network*, vol. 15, p. 549559, 2002.
- [13] P.-Y. Oudeyer, F. Kaplan, and V. Hafner, “Intrinsic motivation for autonomous mental development,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 2, p. 265286, 2007.
- [14] X. Huang and J. Weng, “Inherent value systems for autonomous mental development,” *International Journal of Humanoid Robotics*, vol. 4, no. 2, pp. 407–433, 2007.
- [15] J. L. Krichmar, “The neuromodulatory system: A framework for survival and adaptive behavior in a challenging world,” *Adaptive Behavior*, vol. 16, no. 6, pp. 385–399, 2008.
- [16] S. Singh, R. L. Lewis, A. G. Barto, and J. Sorg, “Intrinsically motivated reinforcement learning: An evolutionary perspective,” *IEEE Trans. Autonomous Mental Development*, vol. 2, no. 2, pp. 70–82, 2010.
- [17] B. Cox and J. Krichmar, “Neuromodulation as a robot controller,” *IEEE Robotics and Automations Magazine*, vol. 16, no. 3, pp. 72 – 80, 2009.
- [18] J. Daly, J. Brown, and J. Weng, “Neuromorphic motivated systems,” in *Proc. Int’l Joint Conference on Neural Networks*, San Jose, CA, July 31 - August 5, 2011, pp. 2917–2914.
- [19] R. E. Neapolitan, *Learning Bayesian Networks*. Upper Saddle River, NJ: Prentice Hall, 2004.
- [20] M. L. Puterman, *Markov Decision Processes*. New York: Wiley, 1994.
- [21] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen, “Autonomous mental development by robots and animals,” *Science*, vol. 291, no. 5504, pp. 599–600, 2001.
- [22] J. Weng, “Three theorems: Brain-like networks logically reason and optimally generalize,” in *Proc. Int’l Joint Conference on Neural Networks*, San Jose, CA, July 31 - August 5, 2011, pp. 2983–2990.
- [23] A. H. Maslow, “A theory of human motivation,” *Psychological Review*, vol. 50, no. 4, pp. 370–396, 1943.
- [24] M. Domjan, *The Principles of Learning and Behavior*, 4th ed. Belmont, California: Brooks/Cole, 1998.

The Ontology of Consciousness

Eugene M. Brooks, M.D.

Providence Hospital
Bloomfield Hills, Michigan, US
Embrooks2@comcast.net

Abstract---As postulated by myself in previous articles, transgenic research has empirically demonstrated that particular proteins are central in the ontology of consciousness. The research results support the contention that consciousness is physical. An explanation is given of the process in which consciousness takes place.

Keywords---Consciousness; protein; unobservability; qualities; physical.

I. EMPIRICAL EVIDENCE

In a previous article [Brooks, 2011-2012, p. 233] ^[1], I stated that particular proteins constitute consciousness and I provided a summary of several empirical research results which demonstrate strong support for the concept. Proteins have extremely diverse properties and the “consciousness proteins” appear to literally have the character of consciousness. The purpose of the present article is to offer further ideational support for the concept.

Within the last eight years, empirical research has serendipitously but most importantly accomplished a breakthrough in demonstrating and in understanding the mechanism of consciousness. The following is a brief summary of the research: Mice are normally dichromatic, having vision only for the colors of red and green. By transplanting a human gene for the color blue into mice, both Onishi et al, [2005, pp. 1145-56] ^[2] and Jacobs et al, [2007, pp. 1723-5] ^[3] have independently enabled the mice to be trichromatic. The research has demonstrated that the mouse brains were able to integrate the new information in making color discriminations. The color information is a form of consciousness which is newly present within the mice.

In a more recent transgenic experiment, Carey et al [2010, pp. 66-71] ^[4] provided for the novel consciousness of a specific scent in a mosquito. This was accomplished by transplanting into the mosquito a gene for an odor receptor molecule which the mosquito lacked due to a mutation. The gene which coded for the scent was obtained from a fruit fly. In other transgenic experiments, Park et al, [2008, pp. 0156-0170] ^[5] allowed a mouse-like animal called a mole-rat, which genetically lacked the capacity for feeling pain, to be aware of the quality. Also, Roska [2010, p. 11] ^[6] led a group which accomplished a genetic transfer into mice, which were blind from a genetic disease, to recover their vision in yellow color.

Despite the empirical research, it is difficult for the uninitiated to understand how it is possible for a protein,

which is a physical entity, to be consciousness. Consciousness, is almost always regarded by theorists to be non-physical or to have a non-physical component (dualism).

I offer that a deeply revised theoretical understanding of consciousness includes the realization that certain physical entities [“consciousness proteins”] have the characters of qualities; that qualities are consciousness from the moment of birth; and that consciousness is experience which cannot be observed. Proteins have been genetically transferred to enable novel forms of consciousness. The variability of proteins, including their susceptibility to alteration by nerve cell impulses, makes them excellent candidates for the constituents of qualities and therefore for consciousness. I postulate that the qualities of light and sound, as well as the experiences of the other qualities, are the unique ontologies of physical entities.

Consciousness does not appear on the surface to be physical largely because it is completely unobservable. It is entirely natural for something which cannot be observed in any manner to be regarded as immaterial. Unlike ordinary physical objects, consciousness is unique in that it can neither be seen nor touched. Consider ghosts and spirits or the soul. If, for the moment, I am allowed the assumption that consciousness is indeed physical, one is nevertheless prompted to ask, “How could consciousness possibly be physical? How does it come about or take place?”

The answer, very briefly with a further explanation to follow, is that “consciousness proteins” are the central elements of qualities. Further, the qualities are combined as phenomenal objects and are projected or referred to the environment [Brooks, 2011-2012, p. 223 ff] ^[7]. Prior to the explication, it will be well to clarify the meaning of “qualities” and to describe the needed Kantian concept of the “noumenon” as well as the meaning of “projection.”

II. QUALITIES

I believe John Locke (1632-1704) correctly described the sources of the contents of the mind. He maintained [1975, p. 105] ^[8] that all ideas come from “SENSATION” or “REFLECTION.” In regard to sensation, he believed that the mind was blank at birth and that items from the external world entered the mind through the senses. Locke wrote [1975, p. 106] ^[9]; Brooks, 2003, p. 142] ^[10]:

“Whence [come]...all the materials of Reason and Knowledge? To this I answer, in one word, From Experience [consciousness]: In that, all our Knowledge is founded....”

Locke applied this statement all inclusively, to external objects as well as to our knowledge of the “internal Operations of our Minds.” The senses supplied the building blocks for the mind. He was certainly indicating that he regarded sensations, which he called “qualities”, to comprise primary elemental forms of mental processes. *The qualities are the characterizing central elements of the sensations* each of which is subjective and unique, entirely different from each other and unlike anything else in nature.

“REFLECTION” was viewed by Locke as a process in addition to the generation of qualities. It was an entirely internal process in which the mind obtains ideas “by reflecting on its own Operations within it self (sic).” We would consider reflection to be a process involving recombination—the mind combines and recombines the arrangement of the original ideas or “building blocks” to develop new structures (phenomenal objects, “representations”) which can then be built hierarchically into still larger structures [Brooks, 1995] ^[11]. According to Locke, sensations are able to enter the mind from “Powers” (energies) which external objects convey to one’s sense receptor organelles. Lamps convey light energy and bells convey sound energy. Thus, the lamps and bells are perceptible to one only by means of the conveyed energies. The same principle applies to all of the senses whether exteroceptive or interoceptive.

III. NOUMENAL OBJECTS (GK. *NOUS*, MIND)

There can be little doubt that Kant, who was born twenty years after the death of Locke, was very familiar with Locke’s writings. Kant’s view of objects seems to have followed logically from the statements of his predecessor. Kant’s idea of noumena, therefore, is quite understandable as based upon the physical and physiological facts as implied by Locke. Locke wrote that, “...Qualities...in truth are nothing in the Objects themselves, ... [except] Powers to produce sensations in us...” [Brooks, 2003, p. 40] ^[12]. Rather than using the term, “Powers”, modern wording might substitute “energies.” The statement by Locke was powerful. Even today his insight is counterintuitive and difficult for many people to accept. About two hundred years after Locke, the physiologist, Johannes Muller, (1801-1858) made a statement which has a similar meaning and which is regarded as seminal. It is paraphrased in a textbook of general psychology [Gleitman, 1981] ^[13] as stating that: “[T]he differences in experienced quality [sensation] are caused not by the differences in the stimuli but by the different nervous structures which these stimuli excite, most likely in centers higher up in the brain.”

In regard to noumenal objects Kant is interpreted [1965, p. 74] ^[14] as saying, “What objects may be in themselves...remains completely unknown to us.” His writings indicate that the energies, which stimulate our senses, arise from objects which are entirely “real” even though our perceptions of them are obtained only from the energies which

the objects emit or reflect.

Brooks [2003, p. 42] ^[15] notes:

Despite our subjective impression to the contrary, one has to think that Kant was correct in stating that what is in the environment is, in a sense, ‘unknowable.’...[A]s we understand him, he meant this in an ultimate sense; not that we cannot know the tree or rock but ... we can know them only as our senses detect them.”

It is well accepted among theorists, that *all qualities are present in the mind rather than in or on external objects*. The objects are noumenal. That is, limited to themselves they have to be literally invisible, inaudible, intangible and so on. We can become conscious of objects only as the qualities they engender in our minds. The perceptions of objects consist of qualities.

Bertrand Russell’s understanding was fairly similar to that of Kant. Russell observed [1927, p. 264] ^[16], “...we know nothing of the intrinsic quality of the physical world...” Also [1997] ^[17], “...common sense leaves us completely in the dark as to the true intrinsic nature of physical objects....”

Brooks indicated in 1993 [pp. 5-6] ^[18] and later stated in [2003, p. 10] ^[19]:

We normally assume the ‘real’ appearance of an object to be just as we see it, yet ... experience tells us that in an important sense this is not so. For example, under an ordinary microscope the appearance of an object can be radically different from what it is to the naked eye. With the advent of the electron microscope, the change in the ‘reality’ becomes even more radical and the atomic and subatomic levels are almost beyond the consideration of appearance.

I consider the concept of noumena to be extremely important if not essential to the understanding of consciousness. Providing credibility to the concept, it is well known that even in so-called “solid” materials the atoms are far apart and the materials are mostly empty space. Searle [2004, pp 46-47] ^[20] makes a statement similar to those of [Brooks, 1993 ^[21]; 2003, p. 10 ^[22]] even though in later statements in the same writing he appears to remain unconvinced since he defends the position of “direct realism” [2004, p. 76] ^[23]:

. . . the world consists almost entirely of physical particles and everything else is in some way an illusion (like colors and tastes) or a surface feature (like solidity and liquidity) that can be reduced to the behavior of physical particles. At the level of molecular structure the table is not really solid.

Searle [2004, p. 261] ^[24] also expresses the concept that objects do not exist as we perceive them:

The scientific account of perception shows how the peripheral nerve endings are stimulated by objects in the world, and how the stimulation of the nerve endings eventually sends signals into the central nervous system and finally to the brain, and how in the brain the whole set of neurobiological processes causes a perceptual experience. But the only actual object of our awareness is that experience in the brain. There is no way we could ever have direct access to the external world. All we can ever have direct access to is the effect that the external world has on our nervous system

I do not consider this view of objects, a view originally expressed by Kant, to mean that objects are immaterial. The energies received from the environment which stimulate our senses and result in qualities, are real energies and arise from real objects. Thus noumenal objects are entirely *real* even though we do not perceive them as they exist in the environment “within themselves.” Science is ignorant of their intrinsic natures.

IV. PROJECTION

A third concept, which is of the utmost importance for the understanding of consciousness, is that of projection. Following the reception of external energies and their processing into objects within the mind, how does it happen that we experience objects to be in the environment? William James [1904] ^[25] considered the question to pose a paradox and asked essentially, “How can objects be in two places at once?” In what amounts to a discussion of projection, James noted:

The whole philosophy of perception from Democritus’ time downwards has been just one long wrangle over the paradox that what is evidently one reality should be in two places at once, both in outer space and in a person’s mind.

We can now resolve the paradox: *The phenomenal objects are projected (referred) to the external environment* [Brooks, 2011-2012, p. 223] ^[26]. I use the word “projection” in the sense in which it is defined in Webster’s dictionary as meaning a “send[ing] forth in one’s thoughts or imagination.” Phenomenal objects are “misunderstood” by the brain as being located at the points in the external environment from which the originating energies arise—from the noumenal objects. Thus the locations of the noumenal objects and the projected phenomenal objects are the same. Our consciousness is projected to a lamp and we interpret the lamp as being the object of which we are conscious. But, to be exact (as well as counterintuitive), we are not conscious of the lamp as it exists in nature. The lamp is imperceptible without intervening effects. We know it only indirectly through energies it conveys and through nerve cell impulses. We

perceive the lamp as being physical. But we do not see the lamp itself. That which we see is a “representation” of sorts. The “representation” is composed of “consciousness proteins.” (Think of the proteins, as nerve cell impulses or simply as “physical entities” if that is more comprehensible.) The perceptions of all objects reside in our brains. The phenomenal objects are understood (projected) as being located in the environment.

One is justified in asking, “If I stub my toe on a rock, is not the pain and swelling in my toe?” The message of the pain is transmitted to the brain via nerve cells while the swelling is perceived via one’s eyes. Both the pain and the swelling are real even though both the rock and the toe are noumenal. Note that the brain has no mechanism for becoming conscious of perceptual objects in their actual locations which are within the brain itself.

The following statement by psychologist Piaget [1963 ^[27]; Brooks, 2003, p. 43 ^[28]; 2007-8, p. 365] ^[29] mentions:

Infants have to learn from experience that an area outside themselves exists and that the objects which they perceive are located in that area. One way infants corroborate the location of objects they perceive is by touching them.

That referral or projection is a learned function is fascinatingly evidenced by babies as they attempt to relate to themselves in a mirror as a separate person. Animals make the same mistake.

Phenomenal objects are regularly projected (referred) to the environment but there is no projection of anything physical. The projection is purely a psychological event and occurs automatically without awareness of the process or of conscious volition. The objects are simply understood in the mind as if they are located in the environment [Brooks, 1993, pp. 17-19 ^[30]; 2007-2008, p. 361-365 ^[31]; 2010-2011, p. 226 ^[32]].

The following is excerpted and paraphrased from [Brooks, 2007-2008, p. 363 ^[33]]: While almost all theories of consciousness fail to comment about projection and some statements which do contain comment express objection, it is interesting that a conception of projection, very similar to the one I expound in the present writing, was expressed in a book by Whitehead in 1925 [p. 54] ^[34]:

The mind in apprehending also experiences sensations which, properly speaking, are projected by the mind alone. These sensations are projected by the mind so as to clothe appropriate bodies in external nature. Thus the bodies are perceived as with the qualities which in reality do not belong to them, qualities which in fact are purely offsprings of the mind.

It is unfortunate that Whitehead’s observation received too little theoretical elaboration or general reception. Ruch [1950] ^[35], in a textbook of physiology, also describes a concept of

projection very similar to mine as follows:

All our sensations are aroused directly in the brain, but in no case are we conscious of this. On the contrary, our sensations are projected either to the exterior of the body or to some peripheral organ in the body, i.e. to the place where experience has taught us that the acting stimulus arises. The exteroceptive sensations are therefore projected exterior to our body. Sound seems to come from the bell, light from the lamp, etc.... An important aspect of sensation which deserves to be called the *law of projection* is that stimulation of a sensory system at any point central to the sense organs gives rise to a sensation which is projected to the periphery and not to the point of stimulation.

Libet [1996] ^[36] mentions an equivalent to the central concept of projection in disagreement with Velmans' "reflexive" theory of consciousness [1993, p. 94] ^[37]:

It seems to me that the reflexive model is simply a special case of what's going on all the time—subjective referral. If you stimulate the somatosensory cortex electrically, you don't feel anything in the brain or head at all, you feel it out in your hand or wherever the representation is of that cortical site. That applies to all sensibilities. There is referral away from the brain to the body parts; there is referral out into space, if the stimulus appears to be coming from there.

That the normal interpretation or understanding of objects as being located at the source of incoming energies is a complex process and should not simply be taken for granted, is indicated by the fact that it does not always occur correctly. In fact, the process of projection is more convincingly exemplified when the projection to the source is incorrect than when the projection is accurate. A very common example of incorrect referral or projection is that which occurs when one looks in a mirror. We are normally so well acquainted with mirror images that we take them for granted and know fully well where the source of the image is located. However, when a second mirror is needed, as when looking at the back of one's head, the reversal of direction tends to make unfamiliar movements very confusing. Other common examples of incorrect localization include the sound from a ventriloquist's dummy, mirages, echoes, visual "floaters", as well as movies and television. In the latter two the sound emanates from speakers at the sides of the screens and not from the mouths of the actors. In another example, we often hear a sound, perhaps of a voice, and look around to see where to refer the sound. An example, which has been well studied by psychologists, is that much of what one tastes and locates in one's mouth, is actually smell, and its source is therefore in

the nose: "The odor of the substance clearly helps [the taste]...as is attested by the common experience that food lacks taste when one has a stuffy head cold" [Vander, Sherman, and Luciano, 1975, p. 531] ^[38].

In the "virtual reality" arrangements used today for training military personnel, one not only perceives objects as being at the supposed origins of the emitted energies but one places oneself in the midst of the scene depicted on a screen. These are all examples in which the mind/brain "mistakenly" interprets stimulated sensory qualities as being located in the environment even though the actual percept is within the brain. Hallucinations and dreams also involve projection. It seems to me that evidence for projection is so commonplace that one hardly needs experimental proof of its occurrence.

V. ONTOLOGY OF CONSCIOUSNESS

The function of projection, in completing the process of perception, provides an answer to James' "paradox." Objects of one's consciousness are actually phenomenal objects which have been projected. The colors, shapes, sizes, etc., (qualities) of which the phenomenal objects consist, are constructions created in the brain and *appear to be the real objects*. The real, noumenal, objects are the sources of the initial stimulations. This status applies to all of subjective "reality", to all objects of perception, whether they are located in the environment external to the body or in the body itself. Because qualitized objects are merely understood to be in the environment, *the objects, which are "clothed" in qualities, are illusional in part*. It is the noumenal objects which are real and present in the environment. The objects which we perceive to be in the environment are projected complexes of qualities [Brooks, 2010-2011, p. 226] ^[39]. We can state that technically "the tree is not green and the sky is not blue." The colors are projected qualities which are located in the mind/brain. The qualities consist of "consciousness proteins" as demonstrated at least in several empirical cases. The proteins [Brooks, 2009-2010 ^[40], 172; 2010-2011, p. 233] ^[41] are the central elements of consciousness.

REFERENCES

- [1] E. M. Brooks, (2011-2012). "Ontology theory of consciousness," in *Imagination, Cognition and Personality*, **31** (3), pp. 217-236.
- [2] A. Onishi et al, (2005). "Generation of knock-in mice carrying third cones with spectral sensitivity different from S and L cones." In *Zoological Science*, **22** (10), pp. 1145-56, Oct.
- [3] G. H. Jacobs, G. Williams, H. Cahill, J. Nathans, (2007), "Emergence of novel color vision in mice engineered to express a human cone pigment," in *Science*, March 23, 2007, Vol. 315, pp. 1723-5.

- [4] A. F. Carey, G. Wang, C. Su, L. S. Zweibel, J. R. Carlson, (2010). "Odorant reception in the malaria mosquito *Anopheles gambiae*," in *Nature*, 464, March 4.
- [5] T. J. Park et al, (2008). "Selective inflammatory pain insensitivity in the African Naked Mole-Rat," in *PLoS Biology*, 6(1).
- [6] B. Roska, (2010). "Gene therapy helps blind mice see," in *Science News*, 178 (2), Jul. 17.
- [7] E. M. Brooks, (2011-2012). "Ontology theory of Consciousness," in *Imagination, Cognition and Personality*, 31 (3), p. 223 ff.
- [8] J.Locke, (1975). *An Essay Concerning Human Understanding*. Peter H. Nidditch (Ed.), Oxford: Clarendon Press, ch. VII, sec. 10, p. 105.
- [9] J.Locke, (1975). *An Essay Concerning Human Understanding*. Peter H. Nidditch (Ed.), Oxford: Clarendon Press, ch. VII, sec. 10, p. 106.
- [10] E. M. Brooks, (2003). *Journey into the Realm of Consciousness: How the brain produces the mind*. Bloomington, IN: First Books, Imprint Author House.
- [11] E. M. Brooks, (1995). "Consciousness and mind: the answer to the hard problem." Unpublished
- [12] E. M. Brooks, (2003). *Journey into the Realm of Consciousness: How the brain produces the mind*. Bloomington, IN: First Books, Imprint Author House.
- [13] H. Gleitman, (1981). *Psychology*. New York: W. W. Norton & Company. p. 288.
- [14] Kant, Immanuel, (1965). *Critique of Pure Reason*. Trans. Smith, Norman, K. Reprinted New York: St. Martin Press.
- [15] E. M. Brooks, (2003). *Journey into the Realm of Consciousness: How the brain produces the mind*. Bloomington, IN: First Books, Imprint Author House. p. 42.
- [16] B. Russell, (1927), *The Analysis of Matter*, New York: Harcourt, Brace & Company, Inc., p. 264.
- [17] B. Russell, (1997), *The Analysis of Matter*, New York: Harcourt, Brace & Company, Inc.
- [18] E. M. Brooks, (1993). "Toward an understanding of perception, consciousness, and meaning." Unpublished.
- [19] E. M. Brooks, (2003). *Journey into the Realm of Consciousness: How the brain produces the mind*. Bloomington, IN: First Books, Imprint Author House. p. 10.
- [20] J. Searle, (2004). *Mind: A Brief Introduction*. Oxford: Oxford University Press.
- [21] E. M. Brooks, (1993). "Toward an understanding of perception, consciousness, and meaning." Unpublished.
- [22] E. M. Brooks, (2003). *Journey into the Realm of Consciousness: How the brain produces the mind*. Bloomington, IN: First Books, Imprint Author House. p. 10.
- [23] E. M. Brooks, (2004-2005) "Multiplicity of consciousness," in *Imagination, Cognition and Personality*, 24 (3), p. 76.
- [24] J. Searle, (2004). *Mind: A Brief Introduction*. Oxford: Oxford University Press.
- [25] W. James, (1950). *The Principles of Psychology*, New York, Dover Publications. (First published 1890 by Henry Holt and company, 29(5), 5230).
- [26] E. M. Brooks, (2011-2012). "Ontology theory of consciousness," in *Imagination, Cognition and Personality*, 31 (3), p. 223
- [27] J. Piaget, (1963). *The Origins of Intelligence in children*, New York, W. W. Norton Company, Inc.
- [28] E. M. Brooks, (2003). *Journey into the Realm of Consciousness: How the brain produces the mind*. Bloomington, IN: First Books, Imprint Author House.
- [29] E. M. Brooks, (2007-8). "The perceptual and Personality, 27 (4), pp. 361-365.
- [30] E. M. Brooks, (1993). "Toward an understanding of perception, consciousness, and meaning." Unpublished.
- [31] E. M. Brooks, (2007-8). "The perceptual arc: Unraveling the mysteries of perception," in *Imagination, Cognition and Personality*, 27 (4), p. 363.
- [32] E. M. Brooks, (2010-11). 'The basis of epistemology: How the brain understands and develops meaning,' in *Imagination, Cognition and Personality*, 30 (2), p. 226.
- [33] E. M. Brooks, (2007-8). "The perceptual arc: Unraveling the mysteries of perception," in *Imagination, Cognition and Personality*, 27 (4), p. 363.
- [34] A. N. Whitehead, (1925). *Science and the Modern World*. New York: The Free Press, a division of the Macmillan Publishing Col, Inc
- [35] J. F. Ruch, (1955). *A Textbook of Physiology*. Seventeenth Ed., J. F. Fulton, editor. Philadelphia: W. B. Saunders Company.
- [36] B. Libet, (1996). *Solutions to the Hard Problem of Consciousness*. in *Journal of Consciousness Studies*. 3 (1), p. 34.
- [37] M. Velmans, (1993). "A reflexive science of consciousness, Experimental and theoretical studies of consciousness," pub. by Ciba, Chichester: Wiley, 174, pp.81-99.
- [38] A. Vander, J. Sherman, D. Luciano, (1975). *Human Physiology—The mechanisms of body function*. New York: McGraw-Hill Inc., 2nd Ed. p. 531.
- [39] E. M. Brooks, (2010-11). 'The basis of epistemology: How the brain understands and develops meaning,' in *Imagination, Cognition and Personality*, 30 (2), p. 226.
- [40] E. M. Brooks, (2009-2010). "Beyond the identity theory," in *Imagination, Cognition and Personality*, 29(2), p. 172.
- [41] E. M. Brooks, (2010-11). 'The basis of epistemology: How the brain understands and develops meaning,' in *Imagination, Cognition and Personality*, 30 (2), p. 233.

Motor neuron splitting for efficient learning in Where-What Network

Zeja Zheng, Kui Qian, Juyang Weng, and Zhengyou Zhang

Abstract—Biologically-inspired developmental Where-What Network gives an elegant approach to the general visual attention-recognition (AR) problem. In their work [1], Luciw and Weng build the visuomotor network for detecting and recognizing objects from complex backgrounds, modeling the dorsal and ventral streams of the biological visual cortex.

Although WWN models the visual cortex to model the attention and segmentation process in visual cortex, the effects of neuromodulator, such as serotonin and dopamine, on individual neurons in the brain are challenging to understand and model, largely because each neuron in an emergent network does not have a static, task-specific meaning. Weng and coworkers modeled the effects of serotonin and dopamine on motor neurons and inner brain neurons in emergent networks as discouragement and encouragement of the firing of neurons, as a statistical effect on the related network behaviors[2].

Directly combining the motivational system with where-what network is plausible but not computationally efficient. The motivational system makes educated guesses for a given foreground object. Where-What Network, on the other hand, requires training in both location motors and type motors. Combining these the two motors will generate a large number of confusing outcomes that takes the network forever to be trained even for a moderate resolution in the location motors.

In this work, we integrate the motivational system with the where-what network based on a coarse to fine learning strategy. Instead of being explicitly informed about the location and type information of the foreground object, which is used in supervised WWN learning, and guessing the correct location and type until correct, which is used in motivated developmental network, the network is rewarded to learn to refine its output on a gradual basis.

The network is first trained to learn rough locations of the foreground object. During the first epoch we train only four different locations: upper left, upper right, lower left and lower right. The network architecture then splits its motor neurons into four exactly same neurons to learn to recognize in higher precision. The new neurons copies the weights and connections of its parent neuron. The four new motor neurons represents four sub-locations of the parent neuron. The network then goes through training process once again to refine those copied neurons. More splitting and training would take place if higher precision is required.

This approach reduces training time thus allows us to train the network efficiently using real time experiment platforms. Experimentally, the recognition rate of the new network is comparable to the original supervised learning network. This approach is also proved to be efficient when applied to type motor.

ACKNOWLEDGEMENT

Zeja Zheng, Kui Qian, and Juyang Weng are with Michigan State University, East Lansing, MI, USA (email zhengzej, qiankui, weng@cse.msu.edu). Juyang Weng is also with the MSU Cognitive Science Program and the MSU Neuroscience Program. Zhengyou Zhang is with Microsoft Research, Redmond, WA, USA. The authors would like to thank Steve Paslaski and Yuekai Wang for their aid in providing the source code for their networks.

REFERENCES

- [1] M. Luciw and J. Weng. Where-what network 3: Developmental top-down attention for multiple foregrounds and complex backgrounds. In *Proc. IEEE International Joint Conference on Neural Networks*, pages 4233–4240, Barcelona, Spain, July 18–23 2010.
- [2] S. Paslaski, C. VanDam, and J. Weng. Modeling dopamine and serotonin systems in a visual recognition network. In *Proc. IEEE International Joint Conference on Neural Networks*, pages 3016–3023, San Jose, CA, July 31 - August 5 2011.

Neuroscientific Critique of Depression as Adaptation

Gonzalo Munevar

Psychology Program, Lawrence Technological University
Southfield, United States
gmunevar@ltu.edu

Donna Irvan

Psychology Program, Lawrence Technological University
Southfield, United States
dirvan@ltu.edu

Abstract—We will discuss evidence from neuroscience against the hypothesis that depression is cognitively adaptive. Andrews and Thomson propose that depression allows for more analytical and focused thinking about our most serious personal problems. It is thus adaptive in a way analogous to disease responses such as fever, which gives an advantage to white cells over pathogens. It is unpleasant but advantageous. Evidence from neuroscience, however, cast doubt on this hypothesis. Some of the key areas involved in the neuroanatomical circuit of depression, such as the prefrontal cortex (reduced volume in the left hemisphere), the dorsal anterior cingulate cortex (decreased activity) and the cortical hippocampal path (disrupted communication), when adversely affected, lead instead to impaired memory and concentration.

Keywords- depression; adaptive; neuroscience; rumination; memory

The analytical rumination hypothesis, proposed by Andrews and Thomson [1], suggests depression is adaptive and evolved as a response to complex problems requiring analytical thought and rumination, which tax the limited processing resources of the individual. The adaptive nature of depression, therefore, minimizes disruption of this process. Creating a lack of interest in activities that normally utilize these limited resources allow the body and mind to monopolize resources to help find resolution. Depression allows the mind to become more focused and analytical, promoting useful cognitive strategies. Since resources are monopolized for analytical problem solving, the biological trade-off for such a process is the depressive mood. Initially this theory does seem to have some plausibility: many behaviors and physiological conditions have proven to be evolutionarily adaptive.

Fever, for example, is an adaptive and beneficial physiological response. Microbial organisms thrive and reproduce rapidly in temperatures comparable to normal human body temperature. When pathogens are introduced, the hypothalamus initiates a rise in temperature, causing the reproduction rate of the pathogen to be severely compromised, while making pathogen-killing phagocytes more active and white blood cells divide quicker [2]. The metabolic cost of such an immunological response is high, often leaving the individual weak and uncomfortable for a period of time. Like the proposed adaptive depression above, fever has a biological trade-off as well.

Unlike the fever account, however, this evolutionary hypothesis for depression does not agree with neurobiological

evidence. At least three key areas of the brain affected by depression lead to impaired memory or concentration. First, data from neuroimaging and postmortem studies provide significant support for a role of the prefrontal cortex (PFC) in major depression. The PFC is necessary for higher cognitive function, and reduced volume in the left hemisphere, caused by cellular atrophy and loss, can lead to impaired concentration [3]. The PFC receives information from many areas in the brain, helps an animal pay attention to the external world, and receives information about internal states [4]. Surely, damage to the PFC places the animal at a disadvantage. The anterior cingulate cortex (AAC) relays information from the limbic area to the PFC, and is crucial in processing attention [4]; thus reduced activation of the AAC, as it happens in depression, can also be expected to result in impaired concentration and diminished, not enhanced, cognitive control. These results alone cast great doubt upon the analytical rumination hypothesis. In addition, the hippocampus and amygdala are important for memory and emotion; depressed individuals have shown reduced volume in the hippocampus, but increased volume in the amygdala [5]. The first leads to even more memory impairments; the second suggests a possible inordinate emphasis on negative emotional memories: more brooding than rumination.

Depression does not appear to be conducive to analytical thinking. It is hard to imagine that reduced concentration, disrupted information relays, and impaired memory would promote any beneficial cognitive processes. It seems unlikely that depression would have evolved as a cognitively adaptive function. Future work may extend to other evolutionary explanations of depression.

REFERENCES

- [1] Paul W. Andrews and J. Anderson Thomson, Jr., "The bright side of being blue: Depression as an adaptation for analyzing complex problems," *Psychological Review* 116(3), 2009. pp. 1-69.
- [2] S. F. Maier and L. R. Watkins, "Cytokines for psychologists: Implications of bidirectional immune-to-brain communication for understanding behavior, mood, and cognition," *Psychological Review* 105(1), 1998. pp. 83-107.
- [3] G. Rajkowska, "Postmortem studies in mood disorders indicate altered numbers of neurons and glial cells," *Biological Psychiatry* 48, 2000. pp. 766-777.
- [4] M. Liotti, and H. S. Mayberg, "The role of functional neuroimaging in the neuropsychology of depression," *Journal of Clinical and Experimental Neuropsychology*, 23, 2001. pp. 121-136.
- [5] J. D. Bremner, M. Narayan, E. R. Anderson, L. H. Staib, H. L. Miller, and D. S. Charney, "Hippocampal volume reduction in major depression," *American Journal of Psychiatry*, 157, 2000. pp. 115-118.

Sponsors

**International Neural Network
Society**



**IEEE Computational
Intelligence Society**



Toyota

TOYOTA

