

Conscious Intelligence Requires Developmental Autonomous Programming For General Purposes

Juyang Weng^{*†‡}

^{*}Department of Computer Science and Engineering,

[†]Cognitive Science Program, [‡] Neuroscience Program,
Michigan State University, East Lansing, MI, 48824 USA

Abstract—Universal Turing Machines are well known in computer science but they are about manual programming for general purposes. Although human children perform *conscious learning* (learning while being conscious) from infancy, it is unknown that Universal Turing Machines can facilitate not only our understanding of Autonomous Programming For General Purposes (APFGP) by machines, but also enable early-age conscious learning. This work reports a new kind of AI—conscious learning AI from a machine’s “baby” time. Instead of arguing what static tasks a conscious machine should be able to do during its “adulthood”, this work suggests that APFGP is a computationally clearer and necessary criterion for us to judge whether a machine is capable of conscious learning so that it can autonomously acquire skills along its “career path”. The results here report new concepts and experimental studies for early vision, audition, natural language understanding, and emotion, with conscious learning capabilities that are absent from traditional AI systems.

I. INTRODUCTION

To be conscious during adulthood, must a system be conscious from its “infancy” all the way into its “adulthood”? All animals [26], [15] conduct lifelong APFGP but traditional AI systems do not. This work argues that consciousness should not be an optional crown jewelry for AI [4], [11] but a necessity for credible AI. Unconscious AI has resulted in machines that are brittle because they are unaware of themselves and the physical world around them. Machines can become highly conscious by bootstrapping its degree of consciousness through lifelong conscious learning.

Consciousness must apply to different contexts that a life experiences. Furthermore, the term involves many entities, such as environment, awareness, cognition and behavior, which is learned through lifetime based on genetically prepositioned (i.e., developed) learning capabilities [18], [17], [4]. For example, how does a cattle or a human in Fig. 1 learn consciousness from infancy so that, after it has grown up, it can navigate autonomously through the hustle and bustle of streets to reach its home?

Can an artificial machine consciously learn to do the same and more? Weng et al. 2001 [25], which started this conference series as ICDL, proposed “autonomous mental development” as learning across lifetime that must be task-nonspecific. However, the link between “autonomous development” and “consciousness” still lacks a computational minimal set.

By “minimal set”, we mean a minimal set of mechanisms from which a machine, natural or artificial, can bootstrap its



Fig. 1. A conscious cattle and conscious humans navigate on a busy street of New Delhi, India.

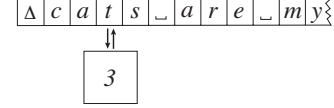


Fig. 2. An example of Turing Machine. Each cell of the tape bears only a symbol. The controller has, at each integer time, a current state (e.g., 3).

degree of consciousness from “early age” to “later age” so that a degree of consciousness is already present during learning, not only after a static batch of learning. This minimal set makes the causality of consciousness clearer.

In order to gain understanding of what that minimal set is in terms of computation, let us start with a model of computation well known in computer science but not directly related to consciousness till this work. Please skip no sections.

II. TURING MACHINES

Turing Machines, originally proposed by Alan Turing [23] in 1936, although not meant to explain consciousness, can assist us to understand how consciousness arises from computations by a machine, both natural and artificial.

A Turing Machine [7], [13], illustrated in Fig. 2, consists of an infinite tape, a read-write head, and a controller. The controller consists of a sequence of moves where each move is a 5-word sentence of the following form:

$$(q, \gamma) \rightarrow (q', \gamma', d)$$

meaning that if the current state is q and the current input that the head senses on the tape is γ , then the machine enters to next state q' , writes γ' onto the tape, and its head moves in direction d (left, right, or stay) but no more than one cell away.

Intuitively speaking, let us consider each symbol in the above 5-word expression as a “word”. Then all such 5-word expressions are “sentences”. Thus, a human-handcrafted “program” is a sequence of such 5-word sentences the Turing Machine must follow in computation. Although such sentences are not a natural language, they are more precise than a natural language.

III. UNIVERSAL TURING MACHINES

How did Turing make the above machine general-purpose? All we need is to augment the meaning of the input on the tape: The tape contains two parts, program P and data x .

In his 1936 paper [23], Turing explained in detail how a Turing Machine can be constructed to emulate program P applied to data x . This new kind of Turing Machines is now called Universal Turing Machines [7], [13]. We called it *universal* because the program P on the tape is open-ended, supplied by any users for any purposes. However, Universal Turing Machines are still not conscious.

To see the link between Universal Turing Machines and consciousness, we must break a series of restrictions in Turing Machines, as explained in the next section.

IV. EIGHT CONDITIONS FOR CONSCIOUSNESS

The eight conditions below were not well known to be necessary for consciousness. However, the APFGP capability in the title requires all of them. That said, they are still insufficient for giving rise to APFGP without the full Developmental Networks (DN) to be discussed in the next section.

To facilitate memorization, let us summarize the eight conditions in eight words: Grounded, Emergent, Natural, Incremental, Skulled, Attentive, Motivated, Abstractive, giving acronym GENISAMA. Let us explain each of them below.

Grounded: Grounded means sensors and effectors of a learner must be directly grounded in the physical world in which the learner lives or operates. IBM Deep Blue, IBM Watson, and AlphaGo are not grounded. Instead, it is humans who synthesize symbols from the physical world, and thus shield machines off from the rich physical game environments, including their human opponents.

Emergent: The signals in the sensors, effectors and all representations inside the “skull” of the learner must emerge automatically through interactions between the learner and the physical world by way of sensors, effectors, and a genome (aka developmental program). A genome is meant to fit the physical world through the entire life, not only for a specific task during the life. For example, fruit flies must do foraging, fighting and mating. Thus, task-specific handcrafting of representation in sensors, effectors, and inside the “skull” is inconsistent to consciousness. This emergence requirement rules out task-specific and handcrafted representations, such as weights duplication

in convolution used by deep learning. Likewise, an artificial genetic algorithm without lifetime learning/development does not have anything to emerge since each individual does not learn/develop in lifetime.

Natural: The learner must use natural sensory and natural motor signals, instead of human hand-synthesized features from sensors or hand-synthesized class labels for effectors, because such symbols and labels are not natural without a human in the loop. For robots, natural signals are those directly available from a sensor (e.g., RGB pixel values from a camera) and raw signals for an effector/actuator. IBM Deep Blue, IBM Watson and AlphaGo all used handcrafted symbols for the board configurations and symbolic labels for game actions. Such symbols are not natural, not directly from cameras and not directly for robot arms.

Incremental: Because the current action from the learner will affect the next input to the learner (e.g., the current action “turn left” allows it to see left view), learning must take place incrementally in time. IBM Deep Blue, IBM Watson and AlphaGo have used a batch learning method: all game configurations are available as a batch for the learner to learn. The learner is not conscious of how it has improved from early mistakes—not self-conscious.

Skulled: The skull closes the brain of the learner so that any teacher’s direct manipulations with the internal brain representations (e.g., twisting internal parameters) are not permitted. For example, how can the brain be aware of what a neurosurgeon did inside the skull?

Attentive: The learner must learn how to attend to various entities in its environment — the body and extra-body environment. The entities include location (where to attend), type (what to attend), scale to attend (e.g., body, face, or nose), as well as abstract concepts that the learner learned in life (e.g., am I doing the right thing?). IBM Deep Blue, IBM Watson and AlphaGo did not seem to think “what am I doing?”.

Motivated: The beautiful logic that a Universal Turing Machine possesses to emulate any valid program does not give rise to consciousness as we know it. By motivation, we mean that the learner must learn motivation based on its intrinsic motives, such as pain avoidance, pleasure seeking, uncertainty awareness, and sensitivity to novelty. A system that is designed to do facial recognition does not have a motive to do things other than facial recognition. IBM Deep Blue, IBM Watson and AlphaGo did not feel happy when they won a game.

Abstractive: Although a shallow definition of consciousness means awareness, full awareness requires a general capability to abstract higher concepts from concrete examples. By higher concepts here we mean those concepts that a normal individual of a species is expected to be able to abstract. Consider movie “Rain Man”: If a kiss by a lady on the lip is sensed only as “wet”, there is a lack of abstraction. A baby cannot abstract love from the first kiss, but a normal human adult is able to. Thus, abstraction requires learning.

With the above eight requirements, we are ready to discuss GENISAMA Universal Turing Machines as super machines capable of conscious learning.

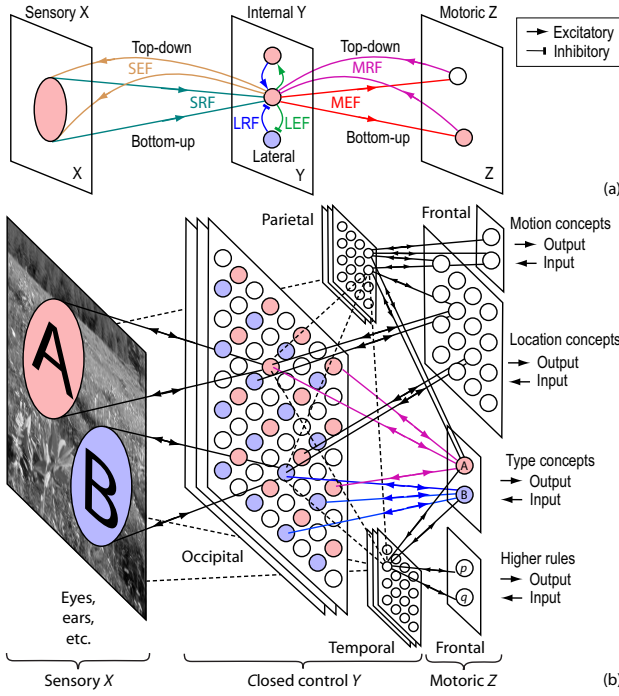


Fig. 3. A DN in (b) grows a brain Y as a two-way bridge between the sensory bank X and the motor bank Z . All the connections are learned, updated and trimmed automatically by DN. (a) A neuron's connections are highly recurrent which requires our time-unfolded explanation.

V. GENISAMA SUPER UNIVERSAL TURING MACHINES

Handcrafting a University Turing Machine is not hard. What is really hard is how to enable such a machine to grow automatically from the natural physical word so as to learn any programs and any data directly from its physical environment! We will see below how.

A DN in Fig. 3 is capable of learning any GENISAMA Universal Turing Machine. It grows one neuron at a time, to learn moves incrementally, one at a time. Such a machine is capable of APFGP, which motivates us as an alternative characterization of consciousness.

A brain is highly recurrent, meaning a neuron sends signals to many other neurons but other neurons also send signals back, directly and indirectly. This recurrence has caused great difficulties in our understanding how the brain works. We must unfold time so that the time-unfolded brain is not recurrent along the time axis. We consider five entities W, Z, Y, X, X' at times $t, t = 0, 1, 2, \dots$, as illustrated in Table I.

The first row in Table I gives the sample times t .

The second row denote the actable world W , such as a hammer acting on a nail.

The third row is the motor Z , which has muscles to drive effectors, such as arms, legs, and mouth.

The fourth row is the skull-closed brain Y . The computation inside the brain must be fully autonomous [25] without pre-given any tasks.

The fifth row is the sensor X , such as cameras, microphones, and touch sensors (e.g., skin).

The last row is the sensible world W' , such as surfaces of objects that reflect light received by cameras.

The actable world W is typically not exactly the same as the sensible world W' , because where sensors sense from and where effectors act on can be different.

Next, we discuss the rules about how a DN, denoted as $N = (X, Y, Z)$, learns from world W and W' .

Extend the tape of the Turing Machine to record images X from sensors, instead of symbols σ . Let X be the original emergent version of input, e.g., a vector that contains values of all pixels.

Extend the output from the Turing Machine (q', γ', d) to be a muscle image from motor Z , instead of symbols. Thus, the GENISAMA Turing Machine directly acts on the physical world.

Parallel computing model: We treat X and Z as external because they can be “supervised” by the physical environment as well as “self-supervised” by the network itself. The internal area Y is closed (hidden)—cannot be directly supervised by external teachers. As in the above Table, we unfold the time t and allow the network to have three areas $X(t)$, $Y(t)$, and $Z(t)$ that learn incrementally through time $t = 0, 1, 2, \dots$:

$$\begin{bmatrix} Z(0) \\ Y(0) \\ X(0) \end{bmatrix} \rightarrow \begin{bmatrix} Z(1) \\ Y(1) \\ X(1) \end{bmatrix} \rightarrow \begin{bmatrix} Z(2) \\ Y(2) \\ X(2) \end{bmatrix} \rightarrow \dots \quad (1)$$

where $\rightarrow$ means neurons on the left adaptively links to the neurons on the right.

Note, all neurons in every column t use only the values of the column $t - 1$ to its immediate left, but use nothing from other columns. This is true for all columns t , with integers $t \geq 1$. Otherwise, iterations are required. Namely, by unfolding time in the above expression, the highly recurrent operations in recurrent DN become nonrecurrent in time-unfolded DN. Thus, the DN runs in real time and do not have to slow down waiting for any iterations.

Using Eq. (1), we outline each of the motor area Z , brain area Y , and sensory area X , for $t = 1, 2, \dots$, from an embryo, all the way to an adult, till possible death.

1. The motor area Z starting from $Z(0)$, represents many muscles signals in a developing body. Muscle cells in $Z(t)$ at time t take inputs from the $Y(t - 1)$ area and the $Z(t - 1)$, acting on the environment at time t using $Z(t)$ through self-supervision—trials, errors, and practices.

2. Concurrently, the brain Y , starting from $Y(0)$, also dynamically develops and grows. Each neuron in $Y(t)$ gets multiple inputs from all three areas, $X(t - 1)$, $Y(t - 1)$ and $Z(t - 1)$. Competition among neurons allows only few Y neurons to win. These winner Y neurons (like an expert team) at the time t column represent the firing of the brain at time t .

3. Likewise, the sensory area X , starting from $X(0)$, also develops within a developing body. What is different between the motor area Z and the sensory area X is that the former develops neurons that drive muscles (and also “feel” the world) but the latter develops receptors that sense the world.

TABLE I
UNFOLDING TIME FOR APFGP IN A DEVELOPMENTAL NETWORK

Time	0	1	2	3	4	5	6	7	...	t
Actable world W	$W(0)$	$W(1)$	$W(2)$	$W(3)$	$W(4)$	$W(5)$	$W(6)$	$W(7)$	...	$W(t)$
Motor Z	$Z(0)$	$Z(1)$	$Z(2)$	$Z(3)$	$Z(4)$	$Z(5)$	$Z(6)$	$Z(7)$	...	$Z(t)$
Skulled brain Y	$Y(0)$	$Y(1)$	$Y(2)$	$Y(3)$	$Y(4)$	$Y(5)$	$Y(6)$	$Y(7)$	...	$Y(t)$
Sensor X	$X(0)$	$X(1)$	$X(2)$	$X(3)$	$X(4)$	$X(5)$	$X(6)$	$X(7)$	...	$X(t)$
Sensible world W'	$W'(0)$	$W'(1)$	$W'(2)$	$W'(3)$	$W'(4)$	$W'(5)$	$W'(6)$	$W'(7)$	...	$W'(t)$

Now, we have the minimal set of mechanisms—Eq. (1) along with above paragraphs 1, 2 and 3—as the *Computational Model of Emergence of Consciousness* for natural and artificial machines to learn consciousness (and consciously learn) throughout lifetime:

As time goes by, the learner looks more and more aware by learning a lifetime program P and lifetime data x from its world, while an optimal GENISAMA Universal Turing Machine emerges as proven mathematically in [24]. Inside the brain this machine autonomously predicts to make a larger, more sophisticated and increasingly integrated program P for all learned purposes that apply to world data x .

In the eyes of humans, this learner becomes increasingly conscious. Although the lifelong-learned consciousness can be extremely complex, the minimal set of computational mechanisms is relatively simple for us to understand, thanks to the Universal Turing Machines.

VI. TECHNICAL DETAIL

A. Developmental Networks for Learning Consciousness

A Developmental Network is meant for consciousness because it is a holistic model for a biological brain, also fully implementable on an artificial machine.

The following section presents Developmental Network 1 (DN-1). Developmental Network 2 (DN-2) is different from DN-1 primarily in the following sense. In DN-1, each of multiple Y areas has a static set of neurons so that the competition within each area is based on a top- k principle. Namely, inhibition among neurons within each area is implicitly modeled by top- k competition.

In the DN-2, however, there is no static assignment of neurons to any regions, so that regions in the DN-2 automatically emerge, along their scales, cascade, and nesting. A major advantage of DN-2 is that a human programmer is not in the loop of deciding the distribution of X-Y-Z mechanisms, relieving human from this intractable task of handcrafting consciousness. A major disadvantage of DN-2 is that its computational explanation is too sophisticated to be included in this paper. Let us leave DN-2 out from this paper and concentrate on DN-1 below.

The hidden Y area corresponds to the entire “brain”. In the following, we assume the brain has a single area Y but it will enable many subareas to emerge.

The brain takes input from vector $(\mathbf{z}, \mathbf{x})$, not just sensory $\mathbf{x}$ but also motor $\mathbf{z}$, to produce an internal response vector $\mathbf{y}$ which represents the best match of $(\mathbf{z}, \mathbf{x})$ with one of many internally stored patterns of $(\mathbf{z}, \mathbf{x})$:

The winner-take-all learning rule, which is highly nonlinear and simulates parallel lateral inhibition in the internal (hidden) area Y is sufficient to prove in [24] that a DN that has sufficient hidden neurons learns any Turing Machine perfectly, immediately, and error-free.

The n neurons in Y give a response vector $\mathbf{y} = (y_1, y_2, \dots, y_n)$ of n neurons in which only the best-matched neuron fires at value 1 and all other neurons do not fire giving value 0:

$$y_j = \begin{cases} 1 & \text{if } j = \underset{1 \leq i \leq n}{\operatorname{argmax}} \{f(\mathbf{t}_i, \mathbf{z}, \mathbf{b}_i, \mathbf{x})\} \\ 0 & \text{otherwise} \end{cases} \quad j = 1, 2, \dots, n, \quad (2)$$

where f is a function that measures the similarity between the top-down weight vector $\mathbf{t}_i$ and the top-down input vector $\mathbf{z}$ [19] as well as the similarity between the bottom-up weight vector $\mathbf{b}_i$ and the bottom-up input vector $\mathbf{x}$. The value of similarity is the inner product of their length-normalized versions [24]. Corresponding to FA, both the top-down weight and the bottom-up weight must match well for f to give a high value as inner product.

The response vector $\mathbf{y}$ the hidden Y area of DN is then used by Z and X areas to predict the next $\mathbf{z}$ and $\mathbf{x}$ respectively in discrete time $t = 1, 2, 3, \dots$:

$$\begin{bmatrix} \mathbf{z}(t-1) \\ \mathbf{x}(t-1) \end{bmatrix} \rightarrow \mathbf{y}(t) \rightarrow \begin{bmatrix} \mathbf{z}(t+1) \\ \mathbf{x}(t+1) \end{bmatrix} \quad (3)$$

where $\rightarrow$ denotes the update on the left side using the left side as input. The first $\rightarrow$ above is highly nonlinear because of the top-1 competition so that only one Y neuron fires (i.e., exactly one component in binary $\mathbf{y}$ is 1). The second $\rightarrow$ consists of simply links from the single firing Y neurons to all firing neurons on the right side.

Like the transition function of a Turing Machine, each prediction of $\mathbf{z}(t+1)$ in Eq. (3) is called a *transition*, but now in real-valued vectors without any symbols. The same $\mathbf{y}(t)$ can also be used to predict the binary (or real-valued) $\mathbf{x}(t+1) \in X$ in Eq. (3). The quality of prediction of $(\mathbf{z}(t+1), \mathbf{x}(t+1))$ depends on how state Z abstracts the external world sensed by X . The more mature the DN is in its “lifetime” learning, the better its predictions.

The expression in Eq. (3), is extremely rich as illustrated in Fig. 3: Self-wiring within a Developmental Network (DN) as

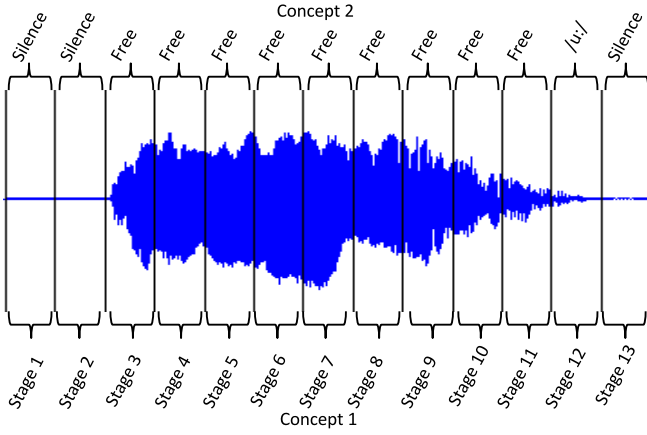


Fig. 5. The sequences of concept 1 (dense, bottom) and concept 2 (sparse, top) for phoneme /u:/. The latest DNs do not need human to provide any labels. Instead, they self-supervise themselves.

scale of attention (global or local), all represented as binary patterns. None is a symbol.

Below, we will see that an auditory consciousness also uses the same DN, but using different “innate” parameters.

Audition from a “lifelong” cochlear sequence: How does a DN become auditory conscious demonstrated by its motor behaviors? Let it learn by artificially “living” in the real world!

For the audition modality, each input image to X is the pattern that simulates the output from an array of hair cells in the cochlea. We model the cochlea in the following way. The cells in the base of the cochlea correspond to filters with a high pass band. The cells in the top correspond to filters with a low pass band. At the same height, cells have different phase shifts. Potentially, such a cochlear model could deal with music and other natural sound, more general than the popular Mel Frequency Cepstral Coefficients (MFCCs) that are mainly for human speech processing. The performance will be reported elsewhere due to the limited space.

Take the phoneme /u:/ as an example shown in Fig. 5. The state of concept 2 keeps as silence when inputs are silence frames. It becomes a “free” state when phoneme frames are coming in, and changes to /u:/ state when first silence frame shows up at the end. At the same time, the states of concept 1 count temporally dense stages.

One may ask, what about higher consciousness such as natural language understanding?

Natural languages from a “lifelong” word sequence: How does a DN become language conscious demonstrated by its motor behaviors? Let it learn by artificially “living” in the real world! Here, we assume grounded words are emergent patterns, not symbols.

As far as we know, this seems to be the first work that deals with language acquisition in a bilingual environment, largely because the DN learns directly from emergent patterns, both in word input and in action input (supervision), instead of static symbols.

The input to X is a 12-bit binary pattern, each represents

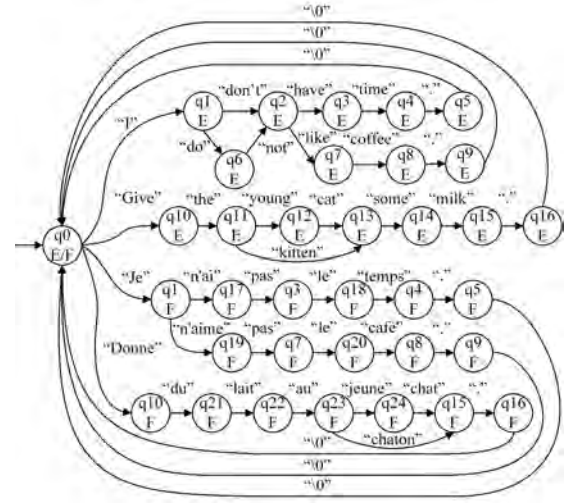


Fig. 6. The finite automaton for the English and French versions of some sentences. The DN learned a much larger finite automaton. Cross-language meanings of partial- and full-sentences are represented by the same state of meaning context q_i , $i = 0, 1, 2, \dots, 24$. See, e.g., q_1 , q_3 , q_4 , and q_5 . But the language specific context is represented by another concept: language type. The last letter is the return character that indicates the end of a sentence.

a word, which potentially can represent 2^{12} words using binary patterns. The system was taught 1,862 English and French sentences from [20], using 2,338 unique words (case sensitive). As an example of the sentences: English: “Christine used to wait for me every evening at the exit.” French: “Christine m’attendait tous les soirs à la sortie.”

The Z area was taught two concepts: language type (English, French, and language neutral, e.g., a number or name) represented by 3 neurons (top-1 firing), and the language-independent meanings as meaning states, as shown in Fig. 6. The latter is represented by 18 neurons (18-bit binary pattern), always top 5 neurons firing, capable of representing $C(18, 5) = 8,568$ possible combinations as states, but only 6,638 actual meanings were recorded. Therefore, the Z area has $3 + 18 = 21$ neurons, potentially capable of representing a huge number 2^{21} binary patterns if all possible binary patterns are allowed.

However, the DN actually observed only 8,333 Z patterns (both concepts combined) from the training experience, and 10,202 distinct (Z, X) patterns—FA transitions. Consider a traditional symbolic FA using a symbolic transition table, which has $6,638 \times 3 = 19,914$ rows and 2,338 columns. This amounts to $19,914 \times 2,338 = 46,558,932$ table entries.

But only $10,202/46,558,932 \approx 0.022\%$ of the entries were detected by the hidden neurons, representing that only 0.02% of the FA transition table was observed and accommodated by the DN. Namely, the DN has a potential to deal with n -tuples of words with a very large n but bounded by DN size, because most un-observed n -tuples are never represented. The FA transition table is extremely large, but never generated.

Without adding noise to the input X , the recognition error is zero, provided that there is a sufficient number of Y neurons. We added Gaussian noise into the bits of X . Let α represent

the relative power of the signal in the noisy signal. When α is 60%, the state recognition rate of DN is around 98%. When α is 90%, the DN has reached 0% error rate, again thanks to the power of DN internal interpolation that converts a huge discrete (symbolic) problem into a considerably smaller continuous (numeric) problem.

Emotional learning using the same network: One may wonder, does this type of consciousness enable emotion? The DN model considers emotion to belong to a wider category known in neuroscience as motivation.

Motivation is very rich [2]. It has two major aspects (a) and (b) in the current DN model. All reinforcement-learning methods other than the DN, as far as we know, are for symbolic methods (e.g., Q-learning [21], [14]) and are in aspect (a) below exclusively. The DN uses concepts (e.g., important events) instead of the rigid time-discount in Q-learning to avoid the failure of far goals.

(a) Pain avoidance and pleasure seeking to speed up learning important events. Signals from pain (aversive) sensors release a special kind of neural transmitters (e.g., serotonin [1]) that diffuse into all neurons that suppress Z firing neurons but speed up the learning rates of the firing Y neurons. Signals from sweet (appetitive) sensors release a special kind of neural transmitters (e.g., dopamine [10]) that diffuse into all neurons that excite Z firing neurons but also speed up the learning rates of the firing Y neurons. Higher pains (e.g., loss of loved ones and jealousy) and higher pleasure (e.g., praises and respects) develop at later ages from lower pains and pleasures, respectively.

(b) Synaptic maintenance—the growing and trimming of the spines of synapses—segments object/event and motivates curiosity. Each synapse incrementally estimates the average error β between the pre-synaptic signal and the synaptic conductance (weight), represented by a kind of neural transmitter (e.g., acetylcholine [27]). Each neuron estimates the average deviation $\bar{\beta}$ as the average across all its synapses. The ratio $\beta/\bar{\beta}$ is the novelty represented by a kind of neural transmitters (e.g., norepinephrine, [27]) at each synapse. The synaptogenic factor $f(\beta, \bar{\beta})$ at each synaptic spine and full synapse enables the spine to grow if the ratio is low (1.0 as default) and to shrink if the ratio is high (1.5 as default). See Fig. 3(b) for how a neuron can cut off their direct connections with Z to become early areas in the occipital lobe or their direct connections with the X areas to become latter areas inside the parietal and temporal lobes. However, we cannot guarantee that such “cut off” are 100% based on the statistics-based wiring theory here.

D. Conclusions

APFGP inside a network has a minimal set of computational mechanisms for conscious systems, natural and artificial. The new APFGP characterization is clearer than existing other characterizations for the notoriously vague term “consciousness” as we discussed in the first section. Hopefully, APFGP will give rise to richer animal-like artificial consciousness so that conscious AI receives a long-overdue

credibility for developmental AI. APFGP might also be useful as a computational model for unifying natural consciousness and artificial consciousness, due to its holistic nature backed by the new capability—APFGP of GENISAMA Universal Turing Machines. Much exciting practical work on learning consciousness remains to be done in the future.

REFERENCES

- [1] N. D. Daw, S. Kakade, and P. Dayan. Opponent interactions between serotonin and dopamine. *Neural Networks*, 15(4-6):603–616, 2002.
- [2] R. J. Dolan. Emotion, cognition, and behavior. *Science*, 298(5596):1191–1194, 2002.
- [3] C. Eliasmith, T. C. Stewart, X. Choo, T. Bekolay, T. DeWolf, Y. Tang, and D. Rasmussen. A large-scale model of the functioning brain. *Science*, 338:1202–1205, 2012.
- [4] J. L. Elman, E. A. Bates, M. H. Johnson, A. Kamiloff-Smith, D. Parisi, and K. Plunkett. *Rethinking Innateness: A connectionist perspective on development*. MIT Press, Cambridge, Massachusetts, 1997.
- [5] C. R. Gallistel. Themes of thought and thinking. *Science*, 285:842–843, 1999.
- [6] E. A. Holm. In defense of the black box. *Science*, 364(6435):26–27, April 5 2019.
- [7] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Boston, MA, 2006.
- [8] R. Jenkins and A. M. Burton. 100% accuracy in automatic face recognition. *Science*, 319(5862):435, 2008.
- [9] M. I. Jordan and T. M. Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349:255–260, July 17 2015.
- [10] S. Kakade and P. Dayan. Dopamine: generalization and bonuses. *Neural Network*, 15:549–559, 2002.
- [11] C. Koch. What is consciousness? *Scientific American*, 318(6):60–64, June 2018.
- [12] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350:1332–1338, 2016.
- [13] J. C. Martin. *Introduction to Languages and the Theory of Computation*. McGraw Hill, Boston, MA, 3rd edition, 2003.
- [14] V. Mnih and et al. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.
- [15] P. R. Montague, P. Dayan, C. Person, and T. J. Sejnowski. Bee foraging in uncertain environments using predictive Hebbian learning. *Nature*, 377:725–728, 1995.
- [16] M. Moravcik and et al. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356:508–513, 2017.
- [17] J. Piaget. *The Construction of Reality in the Child*. Basic Books, New York, 1954.
- [18] J. Piaget and M. Cook. *The Origins of Intelligence in Children*. International Universities Press, Madison, New York, 1952.
- [19] Y. B. Saalman, I. N. Pigarev, and T. R. Vidyasagar. Neural mechanisms of visual attention: How top-down feedback highlights relevant locations. *Science*, 316:1612 – 1615, 2007.
- [20] R. Scriven, G. Amiot-Cadey, and Collins. *Collins French grammar*. HarperCollins, Glasgow, 2011.
- [21] R. S. Sutton and A. Barto. *Reinforcement Learning*. MIT Press, Cambridge, Massachusetts, 1998.
- [22] J. B. Tenenbaum, C. Kemp, T. L. Griffiths, and N. D. Goodman. How to grow a mind: Statistics, structure, and abstraction. *Science*, 331:1279–1285, 2011.
- [23] A. M. Turing. On computable numbers with an application to the Entscheidungsproblem. *Proc. London Math. Soc., 2nd series*, 42:230–265, 1936. A correction, *ibid.*, 43, pp. 544–546.
- [24] J. Weng. Brain as an emergent finite automaton: A theory and three theorems. *International Journal of Intelligent Science*, 5(2):112–131, 2015.
- [25] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen. Autonomous mental development by robots and animals. *Science*, 291(5504):599–600, 2001.
- [26] K. Wynn. Addition and subtraction by human infants. *Nature*, 358:749–750, 1992.
- [27] A. J. Yu and P. Dayan. Uncertainty, neuromodulation, and attention. *Neuron*, 46:681–692, 2005.