

Autonomous Programming for General Purposes: Theory and Experiments

Juyang Weng^{*†‡}, Zejia Zheng^{*†}, Xiang Wu^{**}, Juan Castro-Garcia^{*}

^{*}Department of Computer Science and Engineering,

[†]Cognitive Science Program,

[‡]Neuroscience Program,

Michigan State University, East Lansing, MI, 48824 USA

^{**}School of Automation

Nanjing University of Science and Technology, Nanjing, 210094, China

Abstract—Since the birth of AI, symbols have been well accepted as an abstract representation for intelligent systems, even when neural networks are employed. This theoretical work shows the following new methods: (1) Symbols are probably not used by biological brains because their behaviors are not limited by symbols. (2) Autonomous Programming For General Purposes (APFGP) is necessary for scaling-up AI to animal level intelligence. “Autonomous ” means inside the skull (or network). (3) A Developmental Network (DN) performs APFGP by learning a super Turing machine, called Grounded, Emergent, Natural, Incremental, Skull-closed, Attentive, Motivated, and Abstractive (GENISAMA) Turing machine. (4) A DN is free of any central controller (e.g., Master Map, convolution, or error back-propagation). (5) The GENISAMA DN does APFGP without using symbols. Experiments are reported for vision guided navigation, auditory recognition, and natural language learning. This is the first conference paper on APFGP.

Index Terms—Turing machines, neural networks, autonomous programming, vision, audition, natural language understanding

I. INTRODUCTION

We face not only the challenges of understanding and modeling biological brains but also fundamental limitations in traditional AI, symbolic models and connectionist models alike, due to our use of symbols. It seems to be time for us to *liberate* ourselves from symbols, i.e., free us from the limitation of using symbols.

Plato (427-347 BC) believed largely in nativism. His student Aristotle (384-322 BC), differing from his teacher, argued for three principles of his theory of association—contiguity, frequency, and similarity. Aristotle’s ideas that knowledge emerges from experience have been greatly refined by modern theories of learning in psychology and neuroscience. We humans have made much progress in studying our own brains, but we have not yet seen computational models about how a brain autonomously programs for general purposes. In particular, we often assume that brains use symbols as abstract tokens.

This theoretical work proposes a different view: If the brains must learn APFGP as modeled here, they probably cannot use symbols because the limitations imposed by symbols.

Although often not meant to explain biological learning, artificial intelligence (AI) since Alan Turing 1950 [1] employed

symbols as tokens for concepts and association between concepts. Connectionism (e.g., neural networks) has used some degree of emergent representations, but symbols have also been widely used by connectionists, typically as output tokens from pattern recognition and context tokens.

This theoretical work advocates that outputs, actions, and contexts for AI agents benefit from using patterns that naturally arise, instead of symbols, if AI agents must learn to autonomously program for general purposes like brains can.

After discussing symbols, this theoretical work presents a model for APFGP. Limited by the page limit, this paper provides an introduction suited for a conference, as the first conference paper for APFGP. It does not include all the proofs. The interested reader is referred to Weng arXiv [2] for proofs.

A. Symbols

Symbols (e.g., ASCII codes) are useful for computers. By a set of symbols, we mean a set $S = \{s_1, s_2, \dots, s_n\}$ where symbols s_i and s_j are different if $i \neq j$ and are the same if $i = j$. This “same” or “different” property makes matching any two symbols very simple: success and failure.

To understand the limitations of symbol, we should consider the physical environment in which animals learn. Each animal life consists of body and brain. The brain has two environments, intra-body and extra-body. Suppose that “internal” means inside the skull (or neural network).

A typical symbolic object-to-concept mapping is many-to-one. For example., many real apples correspond to the one abstract class “apple”, or many images of apple 1 correspond to the same object apple 1.

In the following, we will see that the above many-to-one mapping seems unavailable for an animal to learn. The real environments only provide a possibility of learning many-to-many mappings.

B. APFGP

It remains elusive how a biological brain represents, computes, learns, memorizes, updates, and abstracts through its life-long experience—from a zygote, to embryo, fetus, newborn, infancy, childhood, and adulthood. Gradually, the brain

produces behaviors that are increasingly rule-like [3]–[6] and can perform APFGP. By autonomous programming, we mean that a brain autonomously generates a sequence of procedures, from tying shoelaces, to making a business plan, to writing a computer program. Such programs are not just random shufflers. They must relate to meanings of the world — namely physics gives rise to meanings [7], [8].

Here, we greatly simplify such rich processes of co-development of brain and body through lifelong activities, assisted by innate (i.e., prenatally developed) reflexes and innate motivations [9], [10], to realize auto-programming from facts, education, engineering, thinking, fiction, and discovery.

We ask: What is a minimal set of mechanisms that enables a biological or silicon machine to learn APFGP?

In the following, we do not require that symbols are available. We assume the existence of natural patterns that we humans like to consider symbols but they are not.

The remainder of the paper is organized as follows. Section II introduces six requirements that the acronym GENISAMA represents. They seem to be necessary for APFGP. Section III discusses a model for au-programming for general purposes. Section IV provide examples of two tasks and a grand Turing machine for these two tasks as well as the emergent Turing machine. Section V addresses the motivation mechanisms of DN. Section VI provides some discussion and Section VII gives concluding remarks.

II. GENISAMA

Three conceptual steps guide us to reach the theory of APFGP. We first extend Finite Automata (FAs) [11], [12] to agent FA in the sense that states are not hidden but are open as actions. Then we extend agent FAs to attentive agent FAs, so that the machines can automatically attend only a subset of current inputs (e.g., some words among all words on this page). Finally we introduce the GENISAMA Turing Machines by replacing all symbols in such attentive agent FAs with patterns that naturally emerge from the real world.

Agent FA: Two variants of FA, Moore machines and Mealy machines [11], [12] output actions but not their states. We extend an FA to agent [13], called *Agent FA*, by simply requiring it to output its current state entirely, but its current actions are included in the current state. This extension is conceptually important because the current state is now teachable as actions so that we are ready to address the issue of internal representations in neural networks below. In psychology, all skills and knowledge fall into two categories [14], declarative (e.g., verbal) and non-declarative (e.g., bike riding). Therefore, all skills and knowledge can be expressed as actions.

Attentive Agent FA: Suppose that a symbolic street scene at time t has multiple objects. E.g.,

$$S(t) = \{\text{car1, car2, sign1, sign2, pedestrian1, ...}\}$$

Instead of taking only one input symbol σ at a time (e.g., $\sigma = \text{car1}$), an attentive agent FA attends to a set of symbols at a time (e.g., $S'(t) = \{\text{car1, pedestrian1}\} \subset S(t)$). The control

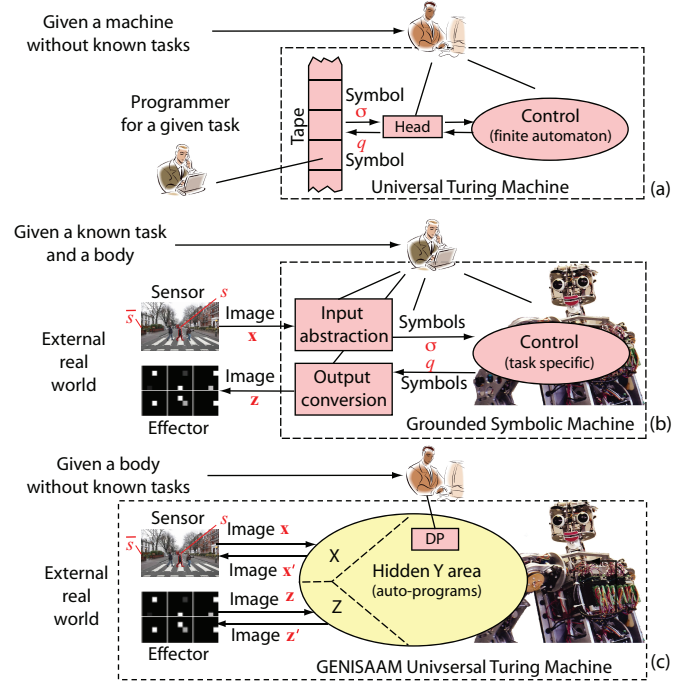


Fig. 1. Three categories of agents, (a) Universal Turing Machines that is symbolic and cannot auto-program, (b) Grounded Symbolic Machines that are task-specific and cannot auto-program, and (c) GENISAMA Universal Turing Machines whose Developmental Program are task-nonspecific. A GENISAMA Universal Turing Machines can auto-program for general purposes. The tape in (a) becomes the real world and all the symbols in (a) and (b) become natural patterns. DP: Developmental Program. X : the sensory port. Z : the effector port. Y : the hidden “bridge for “banks” X and Z . Pink block: human handcrafted. Yellow blocks: emerge automatically.

of any Turing machine (TM) is an Attentive Agent FA as we will discuss below.

Universal TM: In order to understand APFGP, we need to first discuss the Universal TM [11], [12], [15].

Recently, it has been proved by Weng 2015 [16] that the control of any TM is an FA as illustrated in Fig. 1(a). Using this new result, our discussions are much simpler.

Theorem 1: The control of a TM is not only an Agent FA, but also an Attentive Agent FA.

The proof is in [2].

A Universal TM is for general purposes [11], [12]. The input tape of a Universal TM has two parts, the program as instructions and the data for the program to use, not just data like a regular TM. Theorem 1 is also true for any Universal TM because it is a special kind of TM.

Because the input is a set of symbols instead of a symbol, the transition table of an Attentive Agent FA, especially as the control of a Universal TM, is typically extremely large—impractical to handcraft. This consideration supports that machine learning from the real world must be fully automated inside the skull (network).

Next, we drop symbols altogether for our machine. Why? We give three reasons:

1. **Static.** A symbol is static, whose meanings are in the programmer’s document, not told to the symbolic TM. They are also too static for real-time tasks. Suppose you, assisted by a symbolic TM, drive into a new country that uses a new language (e.g., new signs) but the programmer of your symbolic TM has not considered this new language. Your biological brain immediately deals with the *patterns* (e.g., images) of new signs directly without the programmer’s document because you can pull your car over and start to learn. Namely, your brain starts to auto-reprogram itself. But your symbolic TM in Fig. 1(b) cannot because all its symbols are static and your programmer has left you!

2. **Complexity.** Weng [17] proved that your brain is free of symbols for a complexity reason.

3. **Atomic.** A symbol is atomic, having a unique representation, s_i in our above definition. However, when you say, write, or sign an abstract concept, such as the category “apple”, you do not produce exactly the same pattern. For example, your sound waves are different at each different time.

The above discussion prompts us to consider the real physical world when we discuss brain or machine learning, instead in a hypothetical symbolic world.

Therefore, for learning auto-programming, we need a new theory that uses exclusively natural patterns (e.g., image patches of cars and signs from the retina). The acronym GENISAMA below provide six conditions that seem necessary.

GENISAMA TM: As illustrated in Fig. 1(c) it has a Developmental Network (DN) as its control and the real (physical) world as its “tape”. The DN has three areas, sensory X , hidden Y and motoric Z with details shown in Fig. 2. For notational conciseness, we also use X, Y, Z to denote the mathematical vector spaces, respectively, of the corresponding neuronal response patterns denoted as vectors, e.g., $\mathbf{x} \in X$, etc..

If X and Z contain all sensors and effectors of an agent, Y models the entire hidden “brain”. If X and Z correspond to a subpart of the brain areas, Y models the brain area that connect X and Z as a two-way “bridge”. The computational meanings of the acronym GENISAMA are as follows:

Grounded: All patterns $\mathbf{z} \in Z$ and $\mathbf{x} \in X$ are from the external environment (i.e., the body and the extra-body world), not from any symbolic tape.

Emergent: All patterns $\mathbf{z} \in Z$ and $\mathbf{x} \in X$ emerge from activities (e.g., images). All vectors $\mathbf{y} \in Y$ emerge automatically from $\mathbf{z} \in Z$ and $\mathbf{x} \in X$.

Natural: All patterns $\mathbf{z} \in Z$ and $\mathbf{x} \in X$ are natural from real sensors and real effectors, without using any task-specific encoding, as illustrated in Fig. 2.

Incremental: The machine incrementally updates at times $t = 1, 2, \dots$. Namely DN uses $(\mathbf{z}(t), \mathbf{x}(t))$ for update the network and discard it before taking the next $(\mathbf{z}(t + 1), \mathbf{x}(t + 1))$. We avoid storing images for offline batch training (e.g., as in ImageNet) because the next image $\mathbf{x}(t + 1)$ is unavailable without first generating and executing the agent action $\mathbf{z}(t)$ which typically alters the scene that determines $\mathbf{x}(t + 1)$.

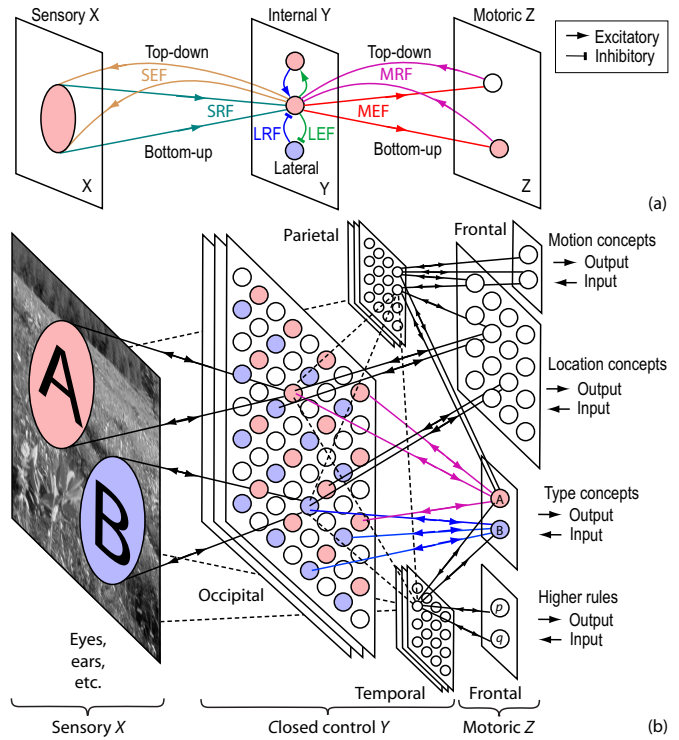


Fig. 2. Brain Y is theoretically modeled as the two-way bridge of the sensory bank X and the motor bank Z . The bridge, is extremely rich: Self-wiring within a Developmental Network (DN) as the control of GENISAMA TM, based on statistics of activities through “lifetime”, without any central controller, Master Map, handcrafted features, and convolution. (a) Each feature neuron has six fields in general. S: Sensory; M: motoric; L: lateral; R: receptive; E: effective; F: field. But simulated neurons in X do not have Sensory Receptive Field (SRF) and Sensory Effective Field (SEF) because they only effect Y and those in Z do not have Motor Receptive Field (MRF) and Motoric Effective Field (MEF) because they only receive from Y . (b) The resulting self-wired architecture of DN with Occipital, Parietal, Temporal, and Frontal lobes. Regulated by a general-purpose Developmental Program (DP), the DN self-wires by “living” in the physical world. The X and Z areas are supervised by physics, including self, teachers, and other physical events. See [2] for why DN does not have any static Brodmann areas.

Skull-closed: As the skull closes the brain to the environment, everything inside the Y area (neurons and connections) are initialized at $t = 0$ and off limit to environment’s direct manipulation after $t = 0$.

Attentive: In every cluttered sensory image $\mathbf{x} \in X$ only the attended parts correspond to the current attended symbol set s . New here is the attention to cluttered motor image $\mathbf{z} \in Z$ so that the attended parts correspond to the current state symbol q (e.g., firing muscle neurons in the mouth and arms). Two symbols correspond to a pattern (not necessarily connected, as in $s = \{\text{car2}, \text{pedestrian1}\}$). Note: The attention here for \mathbf{x} is about the cluttered sensory world, consistent with the literature [18], [19], but the attention in [20], [21] is about the structured internal memory instead inconsistent with the literature.

Motivated: Different neural transmitters have different ef-

fects to different neurons, e.g., resulting in (a) avoiding pains, seeking pleasures and speeding up learning of important events and (b) uncertainty- and novelty-based neuronal connections (synaptic maintenance for auto-wiring) and behaviors (e.g., curiosity).

Abstractive: Each learned concept (e.g., object type) in Z are abstracted from concrete examples in $\mathbf{z} \in Z$ and $\mathbf{x} \in X$, invariant to other concepts learned in Z (e.g., location, scale, and orientation). E.g., the type concept “dog” is invariant to “location” on the retina (dogs are dogs regardless where they are). Invariance is different from correlation: dog-type and dog-location are correlated (e.g., dogs are typically on ground).

DN as the GENISAMA control: Assume a human knowledge base is representable by a grand TM, whose FA control has alphabet $\Sigma = \{\sigma_1, \sigma_2, \dots, \sigma_n\}$, a set of states $Q = \{q_1, q_2, \dots, q_m\}$, and a static lookup table as its transition function $\delta : Q \times \Sigma \mapsto Q$. The lookup table has n columns for n input symbols and m rows for m states. Each transition of the FA control is from state q_i and input σ_j , to the next state q_k , denoted as $(q_i, \sigma_j) \rightarrow q_k$, corresponding to the q_k entry stored at row i and column j , in the lookup table.

Required by GENISAMA, let grounded n (emergent) vectors $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ represent the n (static) symbols in Σ , so that $\mathbf{x}_j \equiv \sigma_j, j = 1, 2, \dots, n$ where \equiv means “corresponds to”. Likewise, let m (emergent) vectors $Z = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_m\}$ represent the m (static) symbols in Q , so that $\mathbf{z}_i \equiv q_i, i = 1, 2, \dots, m$. Thus, each symbolic transition (left, static) in FA corresponds to the vector mapping (right, emergent) in DN:

$$[(q_i, \sigma_j) \rightarrow q_k] \equiv [(\mathbf{z}_i, \mathbf{x}_j) \rightarrow \mathbf{z}_k]. \quad (1)$$

Note that although the right-side vector of the above expression corresponds to a single symbol on the left-side, the symbol on the left-side is only hypothetical but does not seem to exist in the real world (other than in a computer). There are many vectors that a human brain considers to be similar or equivalent.

Because of the complexity reasons in Weng [17], the lookup table for the human common-sense base is exponentially wide and exponentially high, but also extremely sparse. Yet, the right-side in the above Eq.(1) uses only observed sparse entries emerged, where each entry corresponds to a neuron in DN.

In the following, we provide the computation of DN, without taking space to explicitly listing the DN algorithm. The reader is referred to Weng [16] for the DN algorithm.

Denote $\hat{\mathbf{v}} = \mathbf{v}/\|\mathbf{v}\|$, i.e., normalizing the Euclidean length of \mathbf{v} .

The neurons in X and Z are open to the environment, supervised by the environment, or self-supervised by the DN itself, or both. Therefore, we can only assume that X and Z are natural patterns, represented as vectors \mathbf{x} and \mathbf{z} in the mathematical expressions.

Next, let the grand TM in the environment teach the DN by supervising its X and Z ports while TM runs, one transition

at a time in real time. The DN has its brain area Y area hidden (i.e., skulled).

DN Computation: The simplest DN learns incrementally as follows. Given each observation (\mathbf{z}, \mathbf{x}) from the teacher TM, all Y neurons compute their goodness of match. Each Y neuron (i, j) corresponds to an observed transition at the (i, j) entry of the lookup table. In order to match both \mathbf{z} and \mathbf{x} , it has a two-part weights $\mathbf{v}_{ij} = (\mathbf{t}_{ij}, \mathbf{b}_{ij})$. When the best match is not perfect explained below, (\mathbf{z}, \mathbf{x}) is the left-side of a new transition; so DN incrementally adds one more Y neuron by setting its $\mathbf{t}_{ij} = \mathbf{z}_i$ and $\mathbf{b}_{ij} = \mathbf{x}_j$. So, DN adds up to (finite) mn hidden neurons, but typically much fewer because the lookup table is sparse.

The top-down match value is $v_t = \hat{\mathbf{t}} \cdot \hat{\mathbf{z}}$; and bottom-up match $v_b = \hat{\mathbf{b}} \cdot \hat{\mathbf{x}}$. We know that $\hat{\mathbf{a}} \cdot \hat{\mathbf{b}} = \cos \theta$, where θ is the angle between the two unit vectors $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$. $\cos \theta = 1$ is maximized if and only iff $\hat{\mathbf{a}} = \hat{\mathbf{b}}$, namely $\theta = 0$. The match between the current context input (\mathbf{z}, \mathbf{x}) with the weight (\mathbf{t}, \mathbf{b}) of a Y neuron is the sum (or product) of the bottom-up and top-down match values, as its pre-response value:

$$f(\mathbf{z}, \mathbf{x} \mid \mathbf{t}, \mathbf{b}) = v_t + v_b = \hat{\mathbf{t}} \cdot \hat{\mathbf{z}} + \hat{\mathbf{b}} \cdot \hat{\mathbf{x}}$$

Only the best matched Y neuron fires (with response value 1), determined by a highly nonlinear competition:

$$\begin{aligned} (i', j') &= \arg \max_{(i, j) \in Y} f(\mathbf{z}, \mathbf{x} \mid \mathbf{t}_{ij}, \mathbf{b}_{ij}) \\ &= \arg \max_{(i, j) \in Y} \{\hat{\mathbf{t}}_{ij} \cdot \hat{\mathbf{z}} + \hat{\mathbf{b}}_{ij} \cdot \hat{\mathbf{x}}\}. \end{aligned}$$

All other loser Y neurons do not fire (response value 0), because otherwise these neurons not only create more noise but also lose their own long-term memory (since all firing neurons must update using input).

The area Z incrementally updates so that the firing Y neuron (i', j') is linked to all firing components (i.e., 1 not 0) in \mathbf{z}_k , so DN accomplishes every observed transition $(\mathbf{z}_i, \mathbf{x}_j) \rightarrow \mathbf{z}_k$, error-free, as proved in Weng [16].

Using the optimal Hebbian learning in [2], Weng [16] further proved that (1) the weight vector of each Y neuron in the optimal (maximum likelihood) estimate of observed samples in (X, Z) , (2) the weight from each Y neuron (i', j') to each Z neuron k is the probability for (i', j') to fire, conditioned on k fired, and (3) overall, the response vectors \mathbf{y} and \mathbf{z} are both optimal (maximum likelihood).

Thus, DN uses at most mn Y neurons, observes each symbolic transition $(q_i, \sigma_j) \rightarrow q_k$ in TM represented by vector transition $(\mathbf{z}_i, \mathbf{x}_j) \rightarrow \mathbf{z}_k$, and learns each error-free if each input (\mathbf{z}, \mathbf{x}) is noise-free. If input (\mathbf{z}, \mathbf{x}) is noisy, DN is optimal. Namely, DN both “over-fits” and is optimal, regardless input is noisy or noise-free. This is a new proof for TM emerging from DN, shorter but less formal than Weng [16].

Attention corresponds to weights \mathbf{t} and \mathbf{b} partially connected with Z area and X area, respectively, thanks to naturally emerging patterns \mathbf{z} and \mathbf{x} .

III. AUTO-PROGRAMMING

Consider two learning modes. Mode 1: Learn from a teacher TM supervised. Mode 2: Learn from the real physical world without any explicit teacher. For early learning in Mode 1 to be useful for further learning in Mode 2, assume that the patterns in Mode 1 are grounded in (i.e., consistent with) the physical world of Mode 2.

Theorem 2: By learning from any teacher TM (regular or universal) through patterns (Modes 1 and 2) with top-1 firing in Y , the DN control enables a learner GENISAMA TM to emerge inside it with the following properties.

- 1) Sufficient neurons situation: The GENISAMA TM is error-free for all learned TM transitions (Mode 1) and re-substitution of all observed physical experiences (Mode 2).
- 2) Insufficient neurons situation: This happens when the finite n Y neurons have all been activated. The action at time $t+1$ is optimal in the sense of maximum likelihood (but not error-free) in representing the observed context space (z, x) , conditioned on the amount of computational resource n and the experience of learning for all discrete times $0, 1, 2, \dots, t$.

The proof is available in [2].

Next, consider APFGP. We represent each purpose as a TM. Suppose a grand transition table G represents the FA control of a grand TM. This G contains a Universal TM T_u and a finite number of tasks as TMs, $T_i, i = 1, 2, \dots$. Traditionally, T_u is based on a (symbolic) computer language, but here T_u can be in a (non-symbolic) natural language if it is GENISAMA.

Theorem 3: A GENISAMA TM inside DN automatically programs for general purposes $T_i, i = 1, 2, \dots$, after it has learned a Universal TM T_u and the related purposes $T_i, i = 1, 2, \dots$. However, the DN algorithm (developmental program) itself is task-independent and language-independent (e.g., English or Chinese).

The proof is available in [2].

Therefore, it has been constructively proved in mathematics that a machine can perform APFGP, not only a biological machine like a human but also a silicon based robot.

Fig. 3 illustrates a simple DN which does not have Y -to- Y connections. It also compares with symbolic probabilistic network.

Table I compares TMs, Universal TMs, grounded symbolic machines, prior neural networks, and GENISAMA TMs.

IV. EXPERIMENTS

We conducted experiments in which a learning system acts as a Turing Machine that learns one of three well-recognized bottleneck problems in AI, vision, audition and natural language acquisition.

Vision from a “lifelong” retina sequence: In the above discussion, we used vision as an example of the sensory modality. Fig. 5 provides an overview of the extensiveness of the training, regular training, and blind-folded testing sessions. The inputs to the DN were from the same mobile phone

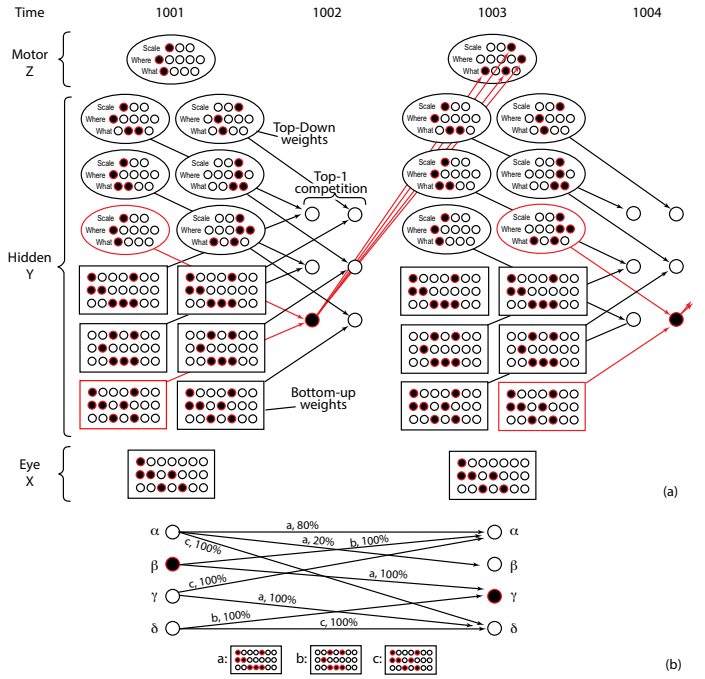


Fig. 3. Architectural contrast between an Emergent DN and a symbolic probabilistic network. (a) Emergent DN. A black circle indicates a firing neuron and a white circle marks a neuron not firing. (b) A symbolic probabilistic network. A black circle indicates an active symbolic state and a white circle marks an inactive symbolic state. Each symbol (α, β, a, b , etc.) in (b) corresponds to a pattern or sub-vector (in Z or X) in (a). Because the representations in ($Z; X$) as patterns in (a) instead of “representationless” symbols in (b), the neural firing values in (a) are based on context (match of patterns and the associated firing values) instead of symbolic yes-no match in (b). Note: both the sensory input vector and the emergent motor vector are typically not exactly the same as their best-matched weight vectors, therefore, an interpolation is allowed for infinitely many possible input vectors. Namely, (a) is “aware” of mismatches in both Z and X , but (b) is only “aware” of mismatch in X in a separate feature detection module but such a mismatch measure is not integrated into the (b) network as (a). For simplicity, the Y area responses at odd time indices and the responses of the X and Z areas at even time indices are not shown.

that performs computation. They include the current image from the monocular camera, the current desirable direction from the Google Map API and the Google Directions API. If the teacher imposes the state in Z , this is treated as the supervised state. Otherwise, the DN outputs its predicted state from Z . The DN learned to attend critical visual information in the current image (e.g., scene type, road features, landmarks, and obstacles) depending on the context of desired direction and the context state. Each state from DN includes heading direction or stop, the location of the attention, and the type of object to be detected (which detects a landmark), and the scale of attention (global or local), as shown on the upper panel of Fig. 4, all represented as binary patterns. None is a symbol.

For further detail of learning vision-guided navigation and planning for navigation see [22].

Below, we discuss two more sensory modalities, audition and natural languages.

Audition from a “lifelong” cochlear sequence: For the audition modality, each input image to X is the pattern that

TABLE I
DIFFERENT TYPES OF MACHINES

Machine types	TM	Universal TM	Grounded symbolic	Prior neural networks	GENISAMA TM
Unknown tasks	No	Yes	No	Pattern recognition only	Yes
General purpose	No	Yes	No	No	Yes
Grounded	No	No	Yes	Yes (can be)	Yes
Auto-program	No	No	No	No	Yes

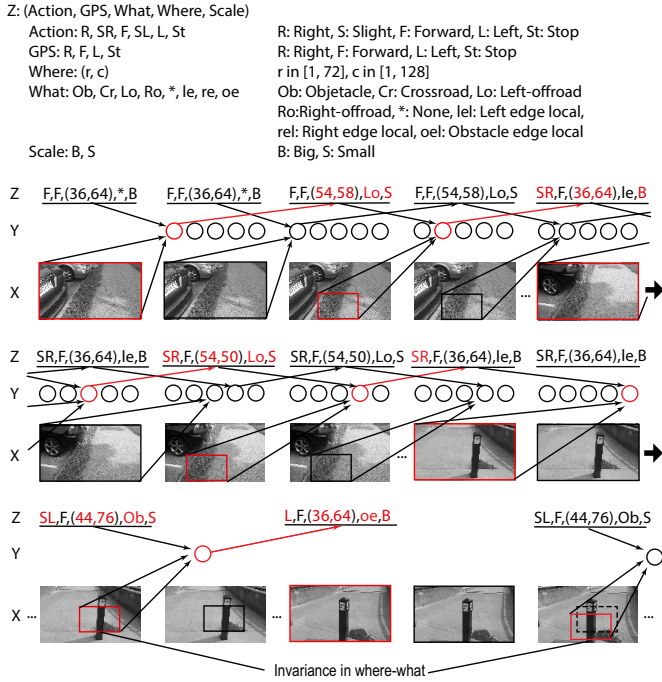


Fig. 4. Examples of how a task-nonspecific and modality-nonspecific engine learns. **Vision modality:** a DN learns concepts like where, what, scale and navigation actions, while learning an attention sequence global, local, global, local ... but any arbitrary attention sequence can be learned in a similar way. The DN has three areas, the sensory area X to take images, the hidden area Y , and the state area Z . The discrete time t increments by 1 from left to right, and continues in the following rows. **Top panel:** A hierarchy of concepts in the Z area incrementally taught by the environment. The Z area has been taught 5 concept zones, Action, GPS, Where, What, Scale. **Lower panels:** auto-wiring in DN to become a detector-recognizer-navigator. Neurons (circles) at each discrete time t only take input from the previous time $t - 1$, $t = 1, 2, \dots$. The discrete time passes from left to right, and continues in the lower panels. Y neurons are generated one at a time before reaching the limit. Z neurons are supervised (teaching) or free (performing) at any time. **Audition modality:** the X area has a firing pattern of the simulated hair cells in the cochlea. **Natural language modality:** the X area has a binary pattern representing a text (word or punctuation).

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.

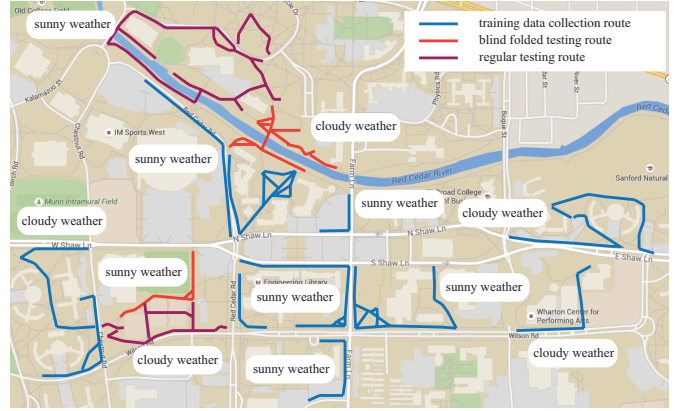


Fig. 5. Training, regular testing, and blind-folded testing sessions conducted on campus of Michigan State University (MSU), under different times of day and different natural lighting conditions (see extensive shadows in Fig. 4). Disjoint testing sessions were conducted along paths that the machine has not learned.

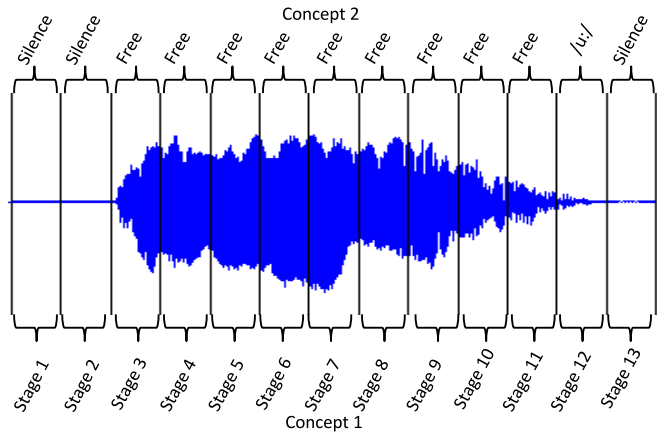


Fig. 6. The sequences of concept 1 (dense, bottom) and concept 2 (sparse, top) for phoneme /u:/.

Take /u:/ as an example shown in Fig. 6. 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.

For more detail of auditory learning using Developmental Networks, see [23].

Natural languages from a “lifelong” word sequence: As far as we know, this seems to be the first work that deals

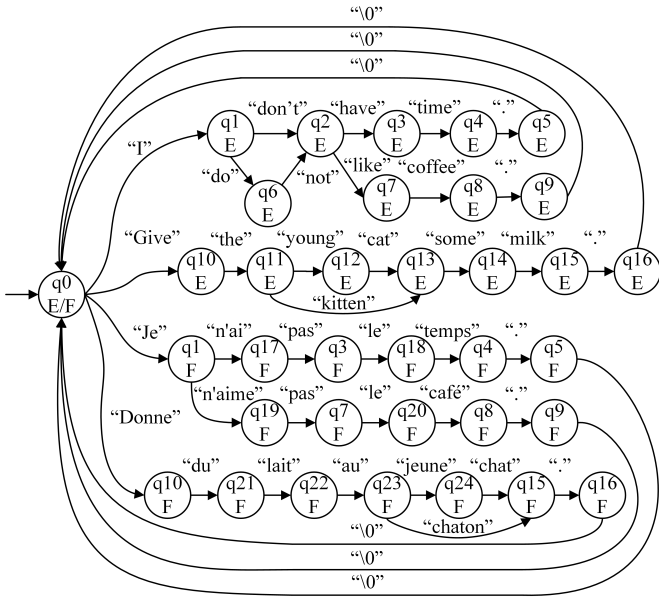


Fig. 7. The finite automaton for the English and French versions of some sentences. The DN learned a much larger finite automaton. Cross-language meanings of partial- and full-sentences are represented by the same state of meaning context q_i , $i = 0, 1, 2, \dots, 24$. See, e.g., q_1 , q_3 , q_4 , and 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.

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 a word, which potentially can represent 2^{12} words using binary patterns. The system was taught 1,862 English and French sentences from [24], 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. 7. 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. See [25] for more detail.

Again, as the only difference from the above two modalities is the patterns in the X area and the Z area, the same DN learns the word inputs and the supervised states.

V. MOTIVATION IN DEVELOPMENTAL NETWORKS

Biological motivation in animals is rich. It has two major aspects (a) and (b) in the current DN model. All reinforcement learning methods other than DN, as far as we know, are for symbolic methods (e.g., Q-learning [26], [27]) and are in aspect (a) exclusively. 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 [28]) 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 [29]) 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 —grows and trims the spines of synapses [30], [31] — to segment object/event and motivate 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 citeYu05). 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 [32]) 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). Each area X , Y , and Z has a prenatal (default) hierarchy of subareas and subsubareas (e.g. Brodmann areas and its subareas for Y) that continuously adapt postnatally. Each area, subarea, subsubarea, has its own synaptogenic factor. This network of synaptogenic factors dynamically organize the complex brain network (e.g., see Felleman & VanEssen [33]). See Fig. 2(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. See [28], [30], [31] for more details about motivational learning in DN.

VI. DISCUSSION

For mathematical simplicity, all the mathematical formulations here do not use the case where Y neurons connect with Y neurons, other than the connection patterns in Fig. 2. Connections from the hidden Y area to the hidden Y area provide functions that a traditional symbolic TM does not have, e.g., better or faster abstraction using sub-action internal representations other than using the final action patterns in the motor area Z only. In DN-1 [31], such Y -to- Y connections use subareas of Y where the number of neurons in each subarea are hand-assigned. In DN-2, this restriction will be removed. An earlier version of this work appeared as a technical report with proofs in arXiv [2].

The lifetime is the sum of all these time segments. The future work includes enabling a robot to fully automatically learn through its lifetime that could potentially last for days, months, years, or decades. We are not there yet.

Because of APFGP, human programmers are relieved from hand-crafting rules of the extremely massive and complex rules in the brain. The hope of Artificial general intelligence (AGI) requires APFGP in order to scale up. It seems that strong AI is possible using the presented theory.

Although theoretically proved and we have demonstrated complex behaviors such as learning planning for navigation [22], rich demonstrations of APFGP are expected as future work for many laboratories from those who learn and adopt of this new kind of AI early.

VII. CONCLUSIONS

The set of mechanisms that enables APFGP corresponds to a GENISAMA Turing machine which emerges from a Developmental Network. The GENISAMA Turing machine is free of any symbols because a pattern corresponding to a symbol, even if it is supervised by a human teacher, does not guarantee that the corresponding emergent pattern in state and action is still unique, which defeats the definition of symbols. This freedom of symbols liberates machines, biological and artificial, to learn fully automatically from its physical world, with or without humans. The three different experimental tasks here used the same DN, indicating the task-nonspecific and modality-nonspecific nature of the new general methods.

REFERENCES

- [1] A. M. Turing. Computing machinery and intelligence. *Mind*, 59:433–460, October 1950.
- [2] J. Weng. A model for auto-programming for general purposes. Technical Report arXiv:1810.05764, Michigan State University, June 13, 2017.
- [3] J. Piaget. *The Construction of Reality in the Child*. Basic Books, New York, 1954.
- [4] M. Cole and S. R. Cole. *The Development of Children*. Freeman, New York, 3rd edition, 1996.
- [5] M. Sur and J. L. R. Rubenstein. Patterning and plasticity of the cerebral cortex. *Science*, 310:805–810, 2005.

- [6] 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.
- [7] S. Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.
- [8] V. Müller. The hard and easy grounding problems. *AMD Newsletter*, 7(1):8–9, 2010.
- [9] C. Blakemore and G. F. Cooper. Development of the brain depends on the visual environment. *Nature*, 228:477–478, Oct. 1970.
- [10] Y. Li, D. Fitzpatrick, and L. E. White. The development of direction selectivity in ferret visual cortex requires early visual experience. *Nature Neuroscience*, 9:676 – 681, 2006.
- [11] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Boston, MA, 2006.
- [12] J. C. Martin. *Introduction to Languages and the Theory of Computation*. McGraw Hill, New York, 4th edition, 2011.
- [13] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Upper Saddle River, New Jersey, 3rd edition, 2010.
- [14] R. Sun, P. Slusarz, and C. Terry. The interaction of the explicit and the implicit in skill learning: A dual-process approach. *Psychological Review*, 112(1):59–192, 2005.
- [15] 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.
- [16] J. Weng. Brain as an emergent finite automaton: A theory and three theorems. *International Journal of Intelligent Science*, 5(2):112–131, 2015. received Nov. 3, 2014 and accepted by Dec. 5, 2014.
- [17] J. Weng. Symbolic models and emergent models: A review. *IEEE Trans. Autonomous Mental Development*, 4(1):29–53, 2012.
- [18] J. Moran and R. Desimone. Selective attention gates visual processing in the extrastriate cortex. *Science*, 229(4715):782–784, 1985.
- [19] B. A. Olshausen, C. H. Anderson, and D. C. Van Essen. A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *Journal of Neuroscience*, 13(11):4700–4719, 1993.
- [20] A. Graves, G. Wayne, and I. Danihelka. Neural Turing machines. Technical report, Google DeepMind, London, UK, Dec. 10, 2014. arXiv:1410.5401.
- [21] A. Graves and et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538:471–476, 2016.
- [22] Z. Zheng, X. Wu, and J. Weng. Emergent neural turing machine and its visual navigation. *Neural Networks*, 110:116–130, 2019.
- [23] X. Wu and J. Weng. The emergent-context emergent-input framework for temporal processing. In *Proc. International Joint Conf. Neural Networks*, pages +1–8, Budapest, Hungary, July 14–19 2019. IEEE Press.
- [24] R. Scriven, G. Amiot-Cadey, and Collins. *Collins French grammar*. HarperCollins, Glasgow, 2011.
- [25] J. Castro-Garcia and J. Weng. Emergent multilingual language acquisition using developmental networks. In *Proc. International Joint Conf. Neural Networks*, pages +1–8, Budapest, Hungary, July 14–19 2019. IEEE Press.
- [26] R. S. Sutton and A. Barto. *Reinforcement Learning*. MIT Press, Cambridge, Massachusetts, 1998.
- [27] V. Mnih and et al. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.
- [28] N. D. Daw, S. Kakade, and P. Dayan. Opponent interactions between serotonin and dopamine. *Neural Networks*, 15(4-6):603–616, 2002.
- [29] S. Kakade and P. Dayan. Dopamine: generalization and bonuses. *Neural Network*, 15:549–559, 2002.
- [30] Y. Wang, X. Wu, and J. Weng. Synapse maintenance in the where-what network. In *Proc. Int'l Joint Conference on Neural Networks*, pages 2823–2829, San Jose, CA, July 31 - August 5, 2011.
- [31] Q. Guo, X. Wu, and J. Weng. Cross-domain and within-domain synaptic maintenance for autonomous development of visual areas. In *Proc. the Fifth Joint IEEE International Conference on Development and Learning and on Epigenetic Robotics*, pages +1–6, Providence, RI, August 13–16 2015.
- [32] A. J. Yu and P. Dayan. Uncertainty, neuromodulation, and attention. *Neuron*, 46:681–692, 2005.
- [33] D. J. Felleman and D. C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1:1–47, 1991.