3D-to-2D-to-3D Conscious Learning

Juyang Weng*†‡§¶
*Department of Computer Science and Engineering, †Cognitive Science Program, ‡Neuroscience Program,
Michigan State University, East Lansing, MI, 48824 USA
§Brain-Mind Institute, Okemos, MI 48864 USA
¶GENISAMA, Okemos, MI 48864 USA

***Abstract*—This is a theoretical paper on conscious learning for thoughts and creativity through general-purpose and autonomous imitation of demonstrations. This conscious learning is end-to-end (3D-to-2D-to-3D) and free from annotations of 2D images and 2D motor images (e.g., a bounding box to be attended to). The conscious learning algorithm directly takes that of the Developmental Networks that has been previously published extensively with rich experimental results. Apparently, humans and animals do this type of fully automated learning daily, but it is unclear a robot can do the same. Recently, [1], [2] presented a theory of conscious learning rooted in emergent universal Turing machines. It appeared to be the first algorithmic level theory of holistic consciousness, other than many papers in the literature about piecemeal consciousness. However, [1], [2] proved only conscious learning in motor-imposed training mode, namely 3D-to-2D taught by 2D motor impositions, free from 2D annotations. This paper fills the challenging gap in [1], [2] so the conscious learning is 3D-to-2D-to-3D (end-to-end) without motor-impositions or computing "inverse kinematics". This is a major departure from traditional AI—handcrafting symbolic labels that tend to be brittle (e.g., for driverless cars) and then "spoon-feeding" pre-collected "big data". Autonomous imitations drastically reduce the teaching complexity compared to pre-collected "big data", especially because no annotations of training data are needed. Furthermore, conscious learning allows creativity beyond what is taught. This work is directly related to consumer electronics because it requires large-scale on-the-fly brainoid chips in future wearable robots/devices for consumers.**

***Index Terms*—Machine learning, on-chip learning, on-the-fly learning, inverse kinematics, brainoid chips, VLSI.**

I. Introduction

This is a theory paper, supported by our published experiments in vision, audition and natural language [3], [4].

There have been many papers about imitation learning [5] but they are all of special purposes, not embedded with an emergent universal Turing machine and are task-specific in the sense it is a human programmer that designs a representation for a given task. Our approach follows the task-nonspecific paradigm in Weng et al. in *Science* 2001 [6]. This theory presents the generality of a new kind of imitation mechanisms for machine thinking [2] and creativity (see Theorem 4), called autonomous imitations without 2D motor-impositions.

Due to the 6-page limitation, this paper cannot cover the neural network DN (other than a sketch) or the 3D-to-2D conscious learning. The reader is referred [3] for the former and [1], [2] for the latter. The major novelty of this paper is to fill the remaining huge gap from 3D-to-2D to 3D-to-2D-to-3D.

Specifically, this work presents a computational and neuromorphic model for conscious learning by autonomous imitations without 2D motor-impositions. Importantly, this model is meant for both robots and humans. The learner observes demonstrations by a 3D physical world, which may include human teachers such as in classroom teaching. He/it observes 3D demonstrations through the 2-D sensors (camera, cochlear, etc.) and autonomously imitates the demonstrations by creating a 2D motor program in muscle arrays. He executes the 2D program back into the 3D world to generate 3D effects. Thus, we call this kind of autonomous, on the fly, end-to-end learning 3D-to-2D-to-3D conscious learning.

Human infants can hardly survive without intensive parent care. However, it is not true that they learn from a blank slate. Typically, the lower the animal species, the more innate behaviors are present in the newborns. We argue that such inborn reflexes are autonomously developed prenatally. E.g., spontaneous retinal signals are required for wiring visual circuits. Such innate behaviors do not need inverse kinematics.

First described by zoologist Konrad Lorenz in the 1930s imprinting occurs when a newly hatched animal (e.g., duckling) forms an attachment to the first moving thing it sees upon hatching. Experiments have shown that imprinting appears to be a quick-learning process—learning the appearance of the first moving object, which is usually the mother.

Human infants do not present imprinting. However, human infants display some innate behaviors too, such as rooting, kicking, and sucking.

Inspired by biological mechanisms of development of brain's motor areas along with the corresponding limbs, developmental robots have two alternatives: (A) Developmental effectors—developing effectors during lifetime, (B) Nondevelopmental effectors—Handcrafting effectors before inception.

Alternative (A) is necessary for those effectors that are so sophisticated that handcrafted effectors do not allow conscious learning to have the required degree of freedom needed by human-level performance. Vocal effectors that make all possible human sounds, not just speech of a pre-specified prosody, are an example of sophisticated effectors. Wu & Weng [7] employed the Candid Covariance-free Incremental (CCI) Principle Component Analysis (PCA) to develop vocal effectors directly from hearing sounds.

Alternative (B) seems to be sufficient for simpler effectors, such as steering, acceleration, and braking, since each effector is one-dimensional and typically changing one effector is sufficient for many cases. This type of effectors can be directly supervised on the motor end, as motor-imposed learning.

In Proc. IEEE 40th International Conference on Consumer Electronics, pages 1-6, Las Vegas, NV, USA, Jan. 7-9. 2022.

Our main goal here is to solve a currently pressing need to address that existing machine learning capabilities are weak, too rigid, and not autonomous.

This paper seems to be the first, as far as the author is aware, on autonomous learning by imitation. This subject goes beyond the current three modes of learning, motor-imposed, reinforcement, and unsupervised. In fact, the internal mechanisms in the Developmental Networks (DNs) used as a supporting learning engine is unsupervised—skull-closed. As we will see below, the new kind of learning—conscious learning by imitation—allows more sophisticated learning subjects, such as sophisticated effectors and internal attention that currently does not have a way to teach.

Thus, conscious learning by imitation is beyond what is practical with the traditional three modes of machine learning.

A DN differs from many well-known neural networks, such as CNNs [8]–[10] and LSTM [11] in many of the following key properties:

Freedom from Post-Selections [12]—picking the luckiest one from multiple networks trained. Well-popularized "deep learning networks" [8]–[11] are incapable of conscious learning due to post-selections and batch learning.

Learning any finite-size Turing machines: error free.

LCA: Use dually optimal LCA (Lobe Component Analysis) [13] neuronal learning, based on Hebbian mechanisms, instead of error-backprop.

Emergent state-input: Learn a state-input transition instead of a input-to-output mapping in the latter, where state and input are emergent vectors to address *the symbol-grounding problem*, also called *the frame problem*.

Emergent hidden areas: All patterns in the hidden area Y emerge from activities.

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.

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

Skull-closed: As the skull closes the brain to the environment, everything inside the hidden Y area (neurons and connections) are initialized at the inception time $t = 0$ 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 winning input. New here is the attention to components in each cluttered sensorimotor frame $(\mathbf{x}, \mathbf{z})$ is automatic from LCA neuronal competition (without manual annotations of $(\mathbf{x}, \mathbf{z})$) where the attended parts correspond to the current winning attention instead of task-specific attentions.

Motivated: Different neural transmitters have different effects 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). Thus lower motivations develop higher motivations, emotions and goals throughout lifetime.

Fig. 1. A setting for a human teacher to teach while kids are autonomous. Picture courtesy of britishcouncil.org.ua.

Abstractive concepts with invariances: 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).

The remainder of the paper is organized as follows. The next section outlines what is conscious learning, for the purpose of self-containedness, but the reader must read [1] first to get the pre-requisite. Then, we discuss the theory of imitation for conscious learning. The analysis of imitation is followed. Finally, we conclude with some remarks.

II. Conscious Learning

A. Definition

Shown in Fig. 1 is a setting of conscious learning. A developmental robot may start from birth and live to over 21 years. Let us define some conditions of conscious learning in computational terms.

Definition 1 (Conscious learning conditions): Conscious learning satisfies the eight (8) properties: GENISAMA (grounded, emergent, natural, incremental, skull-closed, attentive, motivated, abstract), plus two more: (1) life required degree of real-time, (2) conducted by a general-purpose learning engine capable of learning an emergent universal Turing machine.

The animal-like thinking is necessary since consciousness requires thinking. (1) is needed for human sensory refreshing rate. (2) enables the learner to learn any practical concepts and procedures including Autonomous Programming For General Purposes (APFGP) directly from the physical world. Unlike a universal Turing machine, APFGP in DN learns programs directly from the physical world, using the sensorimotor training mode or the autonomous imitation mode.

B. SEB learning modes

Consider Supervised internal representation? Effector imposed? Biased sensors used? We have a new definition of 8 learning modes as SEB learning modes:

Definition 2 (SEB learning modes): Let a text string, seb, be represented by a binary number. s=1: skull-internal representation is partially human supervised, s=0 otherwise; e=1: effectors are imposed, e=0 otherwise; b=1: biased sensors (pain, sweet, instead of unbiased sensors like cameras and microphones) are used; b=0 otherwise. Then, the seb binary codes have 8 patterns, seb=000, seb=001, ... , seb=111.

Therefore, s=1 corresponds to symbolic representations—human crafted task-specific representations, such as SLAM, Markov Decision Process (MDP), Partially Observable MDP, Graphical Models, as well as neural networks that have human handcrafted features such as human selected features in CNN and LSTM. s=0 corresponds to DN and other inside-skull-unsupervised networks (e.g., some reservoir computing?).

The case e=1 means a human teacher imposes effector for teaching purpose.

Note that eb in seb has four binary patterns, eb=11 is a combination of supervised learning and reinforcement learning, which is not common in machine learning publications but allowed.

We are interested in seb=000 during which imitation takes place. seb=010 and seb=001 only occasionally occur like the setting in Fig. 1.

There are some fundamental limitations in current machine-learning methodology fed by static datasets: (1) The non-sensorimotor recursive nature of any datasets. (2) Post-Selections [12], picking the luckiest network without cross-validation. (3) A lack of conscious learning further explained below.

As shown in Fig. 1 or in a driver-less car, the environment is cluttered which contains multiple components. At any time, only relatively few items (e.g., the drawing that the teacher shows in Fig. 1) are related to the current task that needs to be attended to. Typically such related components occupy only a small part of input image. Other components are distractors.

In computer vision, annotation of attended polygons [14] or a rectangle is non-scalable to real-world deployments.

A more promising way is to set the learner free into deployed settings, like two kids and a teacher in Fig. 1, so that the learner learns from his own autonomous actions including internal attentions which are not motor-imposable.

One concern is that the amount of computational power is prohibitive due to the real world complexity. The availability of brain-size and real-time learning chips—brainoid chips—is indeed a current bottleneck.

The DN framework is optimal in the sense of maximum likelihood (ML) (see [15] for a proof for DN-1 and [3] for a proof for DN-2), conditioned on what is called the Three Learning Conditions—(1) learning framework restrictions (e.g., incremental learning and task-nonspecificity), (2) a learning experience, and (3) a limited computational resource. A DN computes the ML-optional emergent Turing machine, which is explainable. In other words, we only need to train one single network for each lifetime training sequence without a need of post-selections.

C. From Turing machines to DN

Turing machines by Alan Turing were not meant to explain conscious learning. But they can assist us to understand how consciousness arises from computations.

A Turing machine 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-tuple of the following form:

$$(q, \gamma) \rightarrow (q', \gamma', d) \tag{1}$$

meaning that if the current state is q and the current input 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).

Weng [15] extended the state space $Q = \{q\}$ to a new form $Q' = \{(q', \gamma', d)\}$. With Q', he proved that the controller of any Turing machine is an agent Finite Automaton (FA), where agent means that the FA outputs its states.

A universal Turing machine reads only an input tape that has two parts, a program and a data set. The program is a sequence of transitions in Eq. (1). The universal Turing machine is designed to emulate the input program on the input data and produces the output to the tape. Because the program can be any procedure in the Church-Turing thesis, it has been widely accepted that the universal Turing machine is a model for general-purpose computers.

Since a DN learns any agent FA ML-optimally, a DN learns any universal Turing machines ML-optimally. See [16] about how to extend the tape of Turing machine to the 3D real world so that a DN learns APFGP.

Running in discrete times, $t = 0, 1, 2, ...$, a DN learns any Turing machines by learning its FA transitions, but in a vector form:

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

where $\rightarrow$ means neurons on the left adaptively links to the neurons on the right. $Z(t)$ is the vector space corresponding to the symbolic space Q' at time t. $X(t)$ is the vector space corresponding to the symbolic input space $\{\gamma\}$. $Y(t)$, absent from the corresponding Turing machine, is the emergent (learned) representation of the skull-closed brain that conducts the interpolations of the vector space mapping from time $t-1$ to time t. Namely the numerical interpolation replaces the rigid look-up table in the traditional Turing machine.

Define $\mathbf{c} = (\mathbf{x}, \mathbf{y}, \mathbf{z}) \in X \times Y \times Z$ as a context. Thus, the transitions in Eq. (2) corresponds to transitions:

$$\mathbf{c}_0 \rightarrow \mathbf{c}_1 \rightarrow \mathbf{c}_2 \rightarrow ... \tag{3}$$

where $\mathbf{c}_t \in X(t) \times Y(t) \times Z(t), t = 0, 1, 2, ...$

At each time t, the physical world provides a sensory image vector $\mathbf{x}_{t-1} \in X(t-1)$; the machine provides a context $(\mathbf{y}_{t-1}, \mathbf{z}_{t-1})$ and its "brain" function f_{t-1} produces a motor vector $\mathbf{z}_t$ and internal response $\mathbf{y}_t$ as $(\mathbf{y}_t, \mathbf{z}_t) =$

$f_{t-1}(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \mathbf{z}_{t-1})$. The motor vector $\mathbf{z}$ could be either taught by a teacher or, more relevant to this work, to start with, randomly selected from a set of "innate" motor vectors during infancy (e.g., cluster vectors in the PCA space of innate vocal tract) and later, self-generated (from an increasingly more mature PCA space).

Unlike symbolic states in a Turing machine, a state as vector $\mathbf{z} \in Z$ emerges autonomously without any humans in the loop of defining and feeding symbols.

The hidden area $Y(t)$ corresponds to the "brain" at time t. It consists of a large number of neurons whose response $\mathbf{y}_t \in Y(t)$ is computed from each neuron's receptive fields in $X(t-1) \times Y(t-1) \times Z(t-1)$.

Learning in Y and Z takes place incrementally in real time so that f_t is different for each t.

In general, the Z area has a number of subareas, each of which may correspond to a limb or a concept which has a number of possible concept values but each time has only 1 concept value. Also, in general, each neuron in Y dynamically learns its competition zone in the context space. Furthermore, the X space is not be shift-invariant either, but $\mathbf{z} \in Z$ learns invariant concepts.

Without loss of generality, we consider below that each of the Y and Z areas uses only a global top-k ($k = 1$) mechanism which self-picks the winner for the entire area.

At time $t = 0$, the life inception takes place. $\mathbf{z}_0$ is supervised at the initial state (e.g., representing initial state "none"). $\mathbf{x}_0$ takes the sensory image at $t = 0$. $\mathbf{y}_0$ is a zero-vector without top-k competition. Each neuron i in Y and Z starts with random weights and firing age $a_i = 0$.

From $t = 1$, the network starts to update forever. Every neuron i in Y and Z computes their match between its weight $\mathbf{w}_i$ and input $\mathbf{c}_i$ as a inner product of two normalized vector $\hat{\mathbf{w}}_i$ and $\hat{\mathbf{c}}_i$:

$$r'_i = \hat{\mathbf{w}}_i \cdot \hat{\mathbf{c}}_i.$$

A perfect match gives $r'_i = 1$. Each area competes by finding the best matching neuron j.

$$j = \arg\max_i \{r'_i\}.$$

The winner j files at $r_j = 1$ and increment its firing age; all other losers $i \neq j$ do not fire and do not increment their firing ages. The winner neuron updates its weight vector using ML-optimal Hebbian rule:

$$\mathbf{w}_j \leftarrow \frac{a_j - 1}{a_j} \hat{\mathbf{w}}_j + \frac{1}{a_j} r_j \hat{\mathbf{c}}_j.$$

The above computes the incremental average of all response-weighted inputs [13].

Why do random weights result in the same network? When the neuron j fires for the first time its age $a_j = 1$, its retention rate $\frac{a_j - 1}{a_j} = 0$ and its learning rate $\frac{1}{a_j} = 1$. The initial random weight vector only effects whether it is the winner but does not affect the updated weight which must be response-weight normalized input $r_j \hat{\mathbf{c}}_j$. Yes, the ML-optimal estimate from the first sample is indeed the input sample! The above expression for the winner leads to the average of response-weighted inputs conditioned on the firing of the neuron, which corresponds to the minimum-variance estimate of response-weighted inputs.

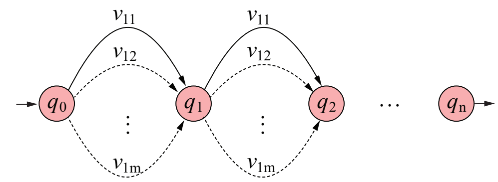

Fig. 2. Abstraction in autonomous imitations of 1 demonstration (solid curves) and $m - 1$ autonomous practices (dashed curves), whose meanings are different from sensorimotor training and typically $m \ll k$ for autonomous imitations in later learning.

Because early age experience is not as important as the latest experience, an amnesic average increases the learning rate $\frac{1}{a_j}$ and accordingly reduces the retention rate so that the sum of them is still 1 [13].

In general, $k > 1$ for top-k competition so that a small percentage of neurons fire each time.

D. Why Autonomous Imitation?

A human senses the 3D world using its sensors whose receptors lie in a 2D sheet (retina, cochlea, skin). For general applicability of our method, we do not need to model the physical transformation from the 3D world to a 2D sensor since our baby brains must work before they have a chance later in life to learn physical laws that govern the mapping from the 3D world to the 2D receptor array. Sometimes, this mapping can be slightly changed, such as wearing a new pair of glasses. But a human can learn quickly and get used to the change. In summary, there is no need to calibrate the transformation from 3D to the 2D.

There are three major reasons to model development of brains in terms of autonomous imitation.

First, imitations are 3D-to-2D-to-3D. A 3D-world event can be a temporal 3D event (e.g., finding how an attended car moves within a time interval), or a combination of space and time (e.g., how a car collision happened). The sensory input to a learner is basically 2D. Autonomous imitations enable a learner to sense a 3D event using its 2D sensors and convert the 2D sensory information into its effectors that generate another but similar 3D event.

Second, autonomous imitations show whether the learner understands the demonstration of a 3D event.

Third, autonomous imitations reduce teaching complexity compared to motor-supervised training as we analyze below.

Let us analyze the imitation complexity. Let a 3D event have n stages. Within each stage, the learner must deal with m variations of stage-to-stage transitions (e.g., due to sensory variations).

Let $n = 10$ and $m = 10$. If we use a brute-force data-fitting network, the learning task requires m^n=$10^{10} = 1$ billion of event samples! Alternatively, if we use motor-imposed training for each stage using human imposed-motor, the same task requires $mn = 10 \times 10 = 100$ teaching examples, 10 teachings for each of the 10 stages. Finally, suppose that the machine is able to autonomously imitate using correct states in contexts, the teacher only needs to demonstrate n stages, one example for each stage. Then, during a later homework session, the learner is able to autonomously imitate for each of the remaining $m - 1 = 9$ variations without a need for the teacher to demonstrate more. Thus, it autonomously generalizes to real-life experience of potentially $m^n = 10^{10} = 1$ billion cases!

Theorem 1 (Imitations reduce teaching complexity): Suppose a task consists of n stages, where each stage consists of dealing with m variations. A bruit force data fitting requires an exponential number $O(m^n)$ training samples and $O(m^n sb)$ computations during training where s is the average receptive field size of neurons and b the number of neurons in the "brain" network. Motor-imposed teaching for an emergent Turing machine in DN requires $O(mn)$ motor-supervision and $O(mnsb)$ computations during training. Autonomous imitation by conscious learning requires $O(n)$ demonstrations and $O(mn)$ autonomous practices as well as $O(mnsb)$ computations during demonstrations and autonomous practices.

Proof: We have already proven above for the training complexity. Let us deal with the number of network weights. Each network update requires $O(sb)$ computations. The number of computations during learning is the number of samples times the number of computations in the network. Thus, we have $O(m^n sb)$ for brute-force data fitting, $O(mnsb)$ for motor-imposed training with abstraction, and $O(nsb)$ for n demonstrations plus $mnsb - nsb = O(mnsb)$ practices through autonomous imitations during homework. ■

The most important concept in the above theorem is the reduction of teaching complexity. Because autonomous imitations directly interact with the real world, they do not need a human teacher to collect a static and large data set and then hand-annotation this data set. Psychologists are amazed by how fast a child learns new sentences without much teaching [17], [18]. Here is a computational account other than "language instinct" [18].

III. Autonomous Imitations

[1] established that using motor-imposed training, a DN ML-optimally learns any grounded Turing machine. If the Turing machine is universal, the DN conducts APFGP. The author argues that APFGP is a computational characterization of consciousness defined in dictionaries.

Let us formally define autonomous imitation.

Definition 3 (Autonomous imitation): A conscious learning agent conducts autonomous imitation using memory learned from its environment if its action sequence imitates a 3D event from the environment and a human expert judges that the action sequence indeed resembles the 3D event. The imitation is autonomous if the agent's effector is not motor-imposed.

Fig. 3 shows an example of autonomous imitation. The 3D event is "A hand places a phone on an ear". The child sees that and her action sequence caused "a hand places a phone on an ear".

Fig. 3. Autonomous imitation. Picture courtesy of Jerry Corley at standupcomedyclinic.com.

Definition 3 does not specify how the 3D event is projected onto the agent's sensors. Neither does it specifies how the agent's effector sequence is judged to resemble the 3D event. Such detail is filled according to the goal of teaching. Definition 3 does not forbid a use of biased sensors to motivate the learner. In animal training, use of reinforcers (e.g., food or touch) is typical.

If the imitation only involves external effectors, motor-imposed teaching is still possible.

However, if the imitation involves skull-internal behavior such as attention (e.g., attention to phone), motor-imposed training is not directly available. A human teacher may use body signs or verbal languages as part of 3D event to facilitate the emergence of imitative behaviors. For example, the teacher could say, "notice the phone" or simply "phone".

IV. Analysis

a) Single-motor imitation: A single motor involves a single segment of the body, such as a vocal tract, a hand, an upper arm, etc. For driverless cars, individual motors include steering, acceleration, braking, etc.

When each Z vector $\mathbf{z}_{\text{innate}}$ is innately firing in the motor, the corresponding physical effect as the corresponding 3D event is simultaneously sensed by the learner's sensors as a sensory event $\mathbf{x}_{\text{effect}}$. After learning $\mathbf{x}_{\text{effect}} \rightarrow \mathbf{z}_{\text{innate}}$, later $\mathbf{z}_{\text{imitate}}$ is invoked from a similar sensory event $\mathbf{x}_{\text{sound}}$ as automatically self-generated $\mathbf{z}_{\text{innate}}$ from $\mathbf{x}_{\text{sound}}$, namely, the "mirror neurons" of $\mathbf{x}_{\text{sound}}$.

Theorem 2 (Early imitation): Early practiced action $\mathbf{z}_{\text{innate}}$ is automatically invoked later from an associated sensory event $\mathbf{x}_{\text{effect}}$:

$$\mathbf{z}_{\text{innate}} \xrightarrow{\text{phy}} \mathbf{x}_{\text{effect}} \xrightarrow{\mathbf{y}} \mathbf{z}_{\text{innate}} \Rightarrow \mathbf{x}_{\text{effect}} \xrightarrow{\mathbf{y}} \mathbf{z}_{\text{innate}} \quad (4)$$

Proof: The proof follows from the above reasoning. In the above expression, "phy" is stands for physics; $\mathbf{y}$ means internal hidden neurons in Y. $\Rightarrow$ means the left side practice causes later autonomous imitation on the right side. ■

b) Multi-motor imitation: A multiple-motor event involves more than a single segment of the body, such as dancing by a humanoid robot and braking while making a turn by a driverless car.

Theorem 3 (Multimotor imitation): A multimotor imitation capability is a later-time extension from the early imitation theorem, by extending the $\mathbf{z}_{\mathrm{innate}}$ to an early practiced multimotor action $\mathbf{z}_{\mathrm{multi}}$ and requiring more fine-tuned neurons $\mathbf{y}_{\mathrm{m}}$ in the neural network that tune their receptive fields to more relevant sensory objects $\mathbf{x}_{\mathrm{multi}}$ that are also sensed from multimotor concepts of the event.

$$\mathbf{z}_{\mathrm{multi}} \xrightarrow{\mathrm{phy}} \mathbf{x}_{\mathrm{multi}} \overset{\mathbf{y}_{\mathrm{m}}}{\to} \mathbf{z}_{\mathrm{multi}} \Rightarrow \mathbf{x}_{\mathrm{multi}} \overset{\mathbf{y}_{\mathrm{m}}}{\to} \mathbf{z}_{\mathrm{multi}} \tag{5}$$

If the autonomous imitation is for a long sequence of event, the above arrows indicate triggering the starting context of the corresponding emergent Turing machines that display the event.

Proof: From Eq. (4), let $\mathbf{z}_{\mathrm{innate}}$ be replaced by $\mathbf{z}_{\mathrm{multi}}$ and $\mathbf{x}_{\mathrm{effect}}$ by $\mathbf{x}_{\mathrm{multi}}$. Assuming that early experience has enabled the neural network to fine turn its hidden feature neurons using Hebbian learning based LCA plus synaptic maintenance by cutting off irrelevant sensory inputs from X and irrelevant concepts inputs from Z. Thus, replacing the symbol $\mathbf{y}$ in Eq. (4) is by $\mathbf{y}_{\mathrm{m}}$, we have the above expression. ■

Theorem 2 can be verbally summarized as "practice makes perfect". For example, to learn how to drive cars one must try driving.

c) Generality and creativity of imitation: In Fig. 3, three concepts are attended to: hand, phone, and ear, and two concept-relationships are attended to, phone-in-hand and phone-at-ear. Two concepts are associated as human type but substituted, "I" substitutes "teacher".

Theorem 4 (Generality and creativity of imitation): Thoughts by a natural or artificial agent via autonomous imitations of 3D real-world events are of general purposes per universal Turing machines. If the imitation result is judged considerably different but creative, such autonomous imitations correspond to creativity of the agent in the judge's eyes.

Proof: Conscious learning in Definition 1 involves learning a universal Turing machine modeled as context transitions in Eq. (3). According to Theorem 3, an imitation composes a program as context transitions, regardless of a computer program or a task plan, which involves attending to some components in contexts, but substituting some associated concepts. According to Eq. (5), this process includes learning to convert a 3D event (e.g., what is taught in a college class) sensed as a sequence of 2D sensory images in the form of $\mathbf{x}_{\mathrm{multi}}$ and then to create a program as a sequence of motor signals in the form of $\mathbf{z}_{\mathrm{multi}}$, and finally to carry out the program back to the real world. Such compositions of programs correspond to human thoughts [2]. Therefore, the context transitions in Theorem 3 are of general purposes per universal Turing machines. The real-world result of the imitated program might not be a 100% duplication of the original 3D event and may be considerably different due to a variety of limitations in the real-world environment and the agent. If the difference is judged by a human expert as creative, the agent is creative in his eyes. ■

Whether an imitation is a children's play or a hypothesis of a scientific principle depends on how experienced the imitator is. The more experienced the imitator is, typically the more valuable the imitation is.

V. Conclusions

This paper has established a general theory of autonomous imitation as (1) learning 3D events, (2) creatively generating a program in 2D motor, and (3) carrying out the program to 3D. The major advance from [1] is that the learner observes teacher's demonstrations using its sensors without motor-imposed training. This is a paradigm shift in AI, addressing the current prevailing problems of Post-Selections in AI. Human-like autonomous learning has become theoretically sound.

This line represents a revolutionarily new direction for future development of consumer electronics—brainoid chips to practically conduct on-the-fly conscious learning at ≈100Hz.

References

[1] J. Weng. Conscious intelligence requires developmental autonomous programming for general purposes. In *Proc. IEEE Int. Conf. on Dev. Learning and Epigenetic Robotics*, pages 1–7, Valparaiso, Chile, Oct. 26-27 2020.

[2] X. Wu and J. Weng. On machine thinking. In *Proc. International Joint Conf. Neural Networks*, pages 1–8, Shenzhen, China, July 18-22 2021. IEEE Press.

[3] J. Weng, Z. Zheng, and X. Wu. Developmental Network Two, its optimality, and emergent Turing machines. U.S. Patent Application Number: 16/265,212, Feb. 1 2019. Approval pending.

[4] X. Wu and J. Weng. Neuron-wise inhibition zones and auditory experiments. *IEEE Transactions on Industrial Electronics*, 66(12):9581–9590, 2019.

[5] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys*, 50:1–35, June 2017.

[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] X. Wu and J. Weng. Muscle vectors as temporally "Dense Labels". In *Proc. International Joint Conference on Neural Networks*, pages 1–8, Glasgow, UK, July 19 - 24 2020.

[8] K. Fukushima, S. Miyake, and T. Ito. Neocognitron: A neural network model for a mechanism of visual pattern recognition. *IEEE Trans. Systems, Man and Cybernetics*, 13(5):826–834, 1983.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1106–1114, 2012.

[10] Y. LeCun, L. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[11] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[12] J. Weng. On post selections using test sets (PSUTS) in AI. In *Proc. International Joint Conference on Neural Networks*, pages 1–8, Shengzhen, China, July 18-22 2021.

[13] J. Weng and M. Luciw. Dually optimal neuronal layers: Lobe component analysis. *IEEE Trans. Autonomous Mental Development*, 1(1):68–85, 2009.

[14] J. Weng, N. Ahuja, and T. S. Huang. Learning recognition and segmentation using the Cresceptron. *International Journal of Computer Vision*, 25(2):109–143, Nov. 1997.

[15] J. Weng. Brain as an emergent finite automaton: A theory and three theorems. *International Journal of Intelligence Science*, 5(2):112–131, 2015.

[16] J. Weng. Autonomous programming for general purposes: Theory. *International Journal of Huamnoid Robotics*, 17(4):1–36, August 2020.

[17] M. Tomasello. The role of joint attentional processes in early language development. *Language Sciences*, 10:69–88, 1988.

[18] S. Pinker. *The Language Instinct: How the Mind Creates Language*. William Morrow, New York, 1994.