An Algorithmic Theory for Conscious Learning

Juyang Weng
juyang.weng@gmail.com
Brain-Mind Institute
Okemos, MI, USA

ABSTRACT

The new conscious learning mode here is end-to-end (3D-to-2D-to-3D) and free from annotations of 2D images and 2D motor images, such as a bounding box for a patch to be attended to. The algorithm directly takes that of the Developmental Networks that has been previously published extensively with rich experimental results. This paper fills the huge gap between 3D world, to 2D sensory images and 2D motor images, back to 3D world so the conscious learning is end-to-end without a need for motor-impositions. This new conscious learning methodology 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". The analysis here establishes that autonomous imitations as presented are a general mechanism in learning universal Turing machines. Autonomous imitations drastically reduce the teaching complexity compared to pre-collected "big data", especially because no annotations of training data are needed. This learning mode is technically supported by a new kind of neural networks called Developmental Network-2 (DN-2) as an algorithmic basis, due to its incremental, non-iterative, on-the-fly learning mode along with the optimality (in the sense of maximum likelihood) in learning emergent super Turing machines from the open-ended real physical world. This work is directly related to electronics engineering because it requires large-scale on-the-fly brainoid chips in conscious learning robots.

CCS CONCEPTS

• Computing methodologies; **• Machine Learning**; **• Learning Paradigms**; **• Unsupervised Learning**;

KEYWORDS

neural networks, conscious machines, sensorimotor learning, inverse kinematics, imitation, universal Turing machines

ACM Reference Format:
Juyang Weng. 2022. An Algorithmic Theory for Conscious Learning. In *AIEE '22: 2022 3rd International Conference on Artirficial Intelligence in Electronics Engineering, Bangkok, Thailand, January 21-23, 2022.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3512826.3512827

AIEE '22, January 21-23, 2022, Bangkok, Thailand
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9548-9/22/01…$XX.00
https://doi.org/10.1145/3512826.3512827

1 INTRODUCTION

This approach follows the task-nonspecific paradigm in Weng et al. in *Science* 2001 [29]. New in this theory is the generality of a new kind of imitation mechanisms for thoughts [35] and creativity (see Theorem 4.3), called autonomous imitations. The algorithmic theory here is supported by our published experiments in vision, audition and natural language [30, 31, 33].

There have been many papers about imitation learning [9] 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.

Due to the 10-page limitation, this paper cannot cover the neural network DN (other than a sketch) or the 3D-to-2D conscious learning. See [25, 35] for the DN-2 algorithm.

Specifically, this work presents a computational and neuromorphic model for conscious learning by autonomous imitations without 2D motor-impositions. 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 the 3D demonstration 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. [2].

First described by zoologist Konrad Lorenz in the 1930s [6], 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. However, this moving thing can also be a human adult, a moving toy, a balloon, or even a stop sign. Imprinting in ducks only occurs during a critical period, starting from 3 hours after hatch, peaked at 15 hours and ends at about 30 hours. Effects of imprinting are lasting, firm, and visually precise.

Human infants do not present imprinting. However, human infants display some innate behaviors too, such as rooting, kicking, and sucking [1]. Infants from 16 to 21 days old imitate (a) tongue protrusion, (b) mouth opening, and (c) lip protrusion demonstrated by an adult [14].

Inspired by biological mechanisms of development of brain's motor areas along with the corresponding limbs, developmental robots have two alternatives for each of its effectors: (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 [34] 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. But conscious learning does not require motor-impositions. For example, in teaching a driverless car, supervised motors receive signals from human manual control. However, to approach human level performance, conscious learning is also highly useful for this type of effectors, because such behaviors depend on attention. For example, a human driver enters a major road from a parking lot is because he has already visually checked that there are no approach vehicles within a considerably long distance, beyond the range of state-of-art ranger sensors. Namely, attention, especially internal attention, as we will define, still has no effective way to motor-impose. Humanoid effectors are also one-dimensional each, but coordination of these one-dimensional effectors is necessary to produce human-like behaviors, such as learning or creation of a new kind of dance.

Our main goal here is to solve a currently pressing need to address that, unlike human learning, existing machine learning capabilities are weak, too rigid, and not autonomous. By weak, we mean that they mainly have motor-imposed mode or reinforcement mode. By too rigid, we mean that they are not applicable to attention, especially language-based attention. By not autonomous, we mean when they learn, what to learn and what to attend, is not autonomously determined by the learner, but are too tedious and slow for a human teacher to tell and annotate in real time.

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 machines do not have a way to be taught while their skulls must be closed.

Thus, conscious learning by imitation seems to be the first algorithmic theory, as far as the author is aware, to practically realize machine consciousness.

A DN differs from many well-known neural networks, such as CNNs [3, 5, 10–12, 19, 32] and LSTM [7] in many of the following key properties:

Freedom from Post-Selections [26] picking the luckiest one from many trained networks.

Learning any finite-size Turing machines: error free.

LCA: Use dually optimal LCA (Lobe Component Analysis) [28] neuronal learning, based on Hebbian mechanisms, instead of error-backprop.

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

Grounded: All patterns $\mathbf{z} \in Z$ and $\mathbf{x} \in X$ are from the external environment, not from any symbolic tape.

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

Natural: All state/motor patterns $\mathbf{z} \in Z$ and sensory patterns $\mathbf{x} \in X$ 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 $(\mathbf{z}(t), \mathbf{x}(t))$, to 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 and any "big data") because of what is called the *sensorimotor recurrence* principle—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 any manual annotations of $(\mathbf{x}, \mathbf{z})$) where the attended parts correspond to the current winning attention instead of task-specific rules (e.g., motion in an object tracking task).

Note: The attention here for $\mathbf{x}$ is about the cluttered sensory world, consistent with the literature[15, 16], but the attention in Graves et al.[4, 5] is about the structured internal memory instead inconsistent with the literature.

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 scaffold higher motivations, emotions and the 5-level consciousness in the Merriam-Webster dictionary.

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, type, 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 at least [25] 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.

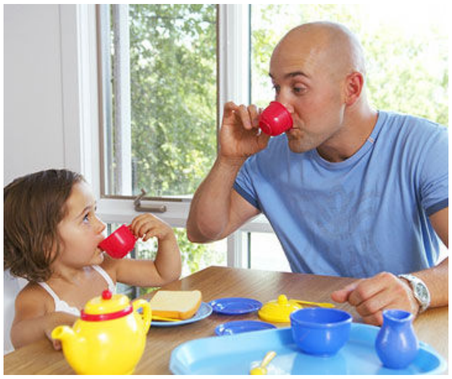

Figure 1: Autonomous imitation without motor imposition. Picture courtesy of thechildrensplaceslt.com.

Rich experimental work supports the feasibility of this new direction, such as the first experimental work that addresses three well-recognized bottleneck problems in AI, vision, audition, and natural language understanding. However, a presentation of experimental support is beyond the scope of this paper.

2 CONSCIOUS LEARNING BACKGROUND

2.1 Definition

Shown in Fig. 1 is a setting of conscious learning in a cluttered scene. A developmental robot may start from birth and live to over 90 years. Let us first define some conditions of conscious learning in computational terms.

Definition 2.1 (Conscious learning conditions). Conscious learning satisfies the eight (8) properties: GENISAMA (grounded, emergent, natural, incremental, skull-closed, attentive, motivated, abstract), plus two more: (1) real-time, on the fly learning, (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 here. See Weng 2020 [24] for APFGP.

2.2 Any combinations of learning modes

Consider three logic bits (seb)—supervised internal representation, effector imposed, biased sensors used. We have a new definition of 8 learning modes as SEB learning modes:

Definition 2.2 (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 since such a case may occur in human learning.

In this work, 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 methodology of machine learning fed by static datasets: (1) The non-sensorimotor recursive nature of any datasets. (2) Post-Selections [26], 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 cup 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 the retinal images. Other components in the retinal images are distractors. In a DN, hidden neurons that win to fire represent attention.

In computer vision, the currently prevailing manual annotations of attended polygons [27] (or a bounding box) in each input image [18] is non-scalable to real-world deployments.

A more promising way is to set the learner free into deployed settings, like that in Fig. 1, so that the learner learns from his own actions including attention actions. It is worth noting that attention is not supervisable because it takes place in hidden neurons inside a closed skull. In conscious learning, attention is self-taught through a process of trial and error.

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 in electronics engineering.

The DN framework is optimal in the sense of maximum likelihood (ML) (see [23] for a proof for DN-1 and [30] 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-optimal emergent Turing machine, which is fully explainable. In other words, we only need to train one single network for each lifetime training sequence without a need of post-selections [26].

2.3 DN optimally learns any grand universal Turing machine from 3D world

Turing machines proposed by Alan Turing [22] were not meant to explain conscious learning. But they can assist us to understand how consciousness arises from computations.

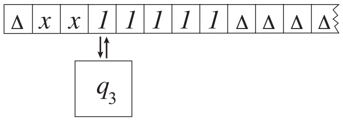

Figure 2: An example of Turing machines. Each cell of the tape bears only a symbol. The controller has, at each integer time, a current state (e.g., q_3).

A Turing machine [8, 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-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 [23] 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. It has been widely accepted that a Turing machine can simulate a procedure of pen-and-pencil. This assumption is called Church-Turing thesis that has been proven by Weng 2020 [24].

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 [24] about how to extend the tape of Turing machine to the 3D real world so that a DN learns APFGP.

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

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

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:

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

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

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. This is the most fundamental reason for fully autonomous learning so that the machine can become increasingly aware through its own interactions with the physical environment. Therefore, area $Z(t)$ takes input from $Y(t-1) \times Z(t-1)$ and its PCA space becomes more and more sophisticated from its "living" experience, probably beyond all the subjects that a programmer has learned.

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 fires 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 [28].

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

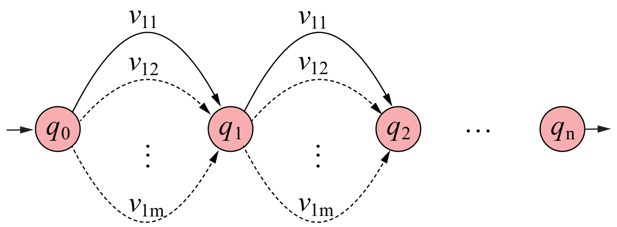

Figure 3: Abstraction in autonomous imitations of 1 demonstration (solid curves) and $m - 1$ autonomous practices (dashed curves), whose meanings are different from sensorimotor training and typically $m \ll k$ for autonomous imitations in later learning.

$\frac{a_j-1}{a_j} = 0$ 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.

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 [28].

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

2.4 Autonomous imitations enable robots to learn like humans: Free from "big data"

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, as illustrated in Fig. 3. 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 2.3 (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-annotate this data set. Psychologists are amazed by how fast a child learns new sentences without much teaching [17, 20, 21]. Theorem 2.3 is a computational account other than "language instinct" [17].

3 CONSCIOUS LEARNING ALGORITHM VIA AUTONOMOUS IMITATION

Weng [25] 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 as defined in dictionaries. Let us formally define autonomous imitation.

Definition 3.1 (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 original 3D event to some degree. The imitation is autonomous if the agent's effector is not motor-imposed.

Fig. 1 shows an example of autonomous imitation of drinking. Specifically, the 3D event to learn is "a hand places a cup on a mouth". The child sees that and her action sequence caused "a hand places a cup on a mouth".

Definition 3.1 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 3.1 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 cup), 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 cup" or simply "cup".

Conscious Learning Algorithm:

Prenatal: Run DN-2 for prenatal training with motor-imposition for "innate" behaviors.

Postnatal: Run DN-2 for postnatal learning without motor-imposition or annotation.

Like a human baby, the algorithm does not require what to learn, as long as the robot lives in a 3D world! Below, we discuss how innate behaviors trigger simple-to-complex imitations.

4 ANALYSIS OF THE CONSCIOUS LEARNING ALGORITHM

A conventional AI method, widely practiced in computer vision, is to hand-label every concept required. For the "where" concept. the label would be a pair (i, j) for i-th column and j-th column in every image. For the "what" concept, the label would be a class label, such as "hand, cup, and mouth". These labels are imposed to the motor area of the agent. This type of training, called motor-imposed training with seb=010, is avoided by autonomous imitations.

The power of conscious learning is rooted in the methodology to set the machine learner free—let it freely act like a human child. For example, during early infancy, the motor area of the machine is driven by a set of innate-behavior vectors.

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, $\mathbf{z}_{\text{innate}}$ becomes the "mirror neurons" of $\mathbf{x}_{\text{sound}}$.

Theorem 4.1 (Early imitation). *Early practiced action $\mathbf{z}_{\text{innate}}$ is automatically invoked later from an associated sensory event $\mathbf{x}_{\text{effect}}$:*

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

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

Note, the sensory event $\mathbf{x}_{\text{effect}}$ on the right side can be from another person. For example, when an infant A innately cries right after birth, the firing motor neuron $\mathbf{z}_{\text{innate}} = \mathbf{z}_{\text{Acry}}$ is the motor of crying sounds of A. When another infant B cries, the similar crying sounds of B, with $\mathbf{x}_{\text{Bcry}} \approx \mathbf{x}_{\text{effect}}$, are sensed by A that causes A's crying. That is, after hearing other baby crying, infant A also cries—autonomous imitation.

The A's crying is not necessarily exactly the same as B. Through later experience of many imitations, refined generalizations in the brain take place as context transitions based on learned emergent Turing machine. These generalizations are increasingly invoked by attended sensory features and motor concepts and become more sophisticated to enable imitations to involve multiple sensory objects and multiple body parts/concepts.

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.

A human teacher demonstrates a multi-motor event, such as dancing. Each body component of the teacher has individual mirror neurons established above. The sensory effects of every body component are sensed by the neural network to excite the corresponding mirror neurons. Thus, attention to multiple sensory components takes place by the firing of multiple motor neurons, either through sequential attention or concurrent joint attention. Multi-motor imitations emerge automatically due to pattern learning.

Theorem 4.2 (Multimotor imitation). *A multimotor imitation capability is a later-time extension from the early imitation theorem, by extending the $\mathbf{z}_{\text{innate}}$ to an early practiced multimotor action $\mathbf{z}_{\text{multi}}$ and requiring more fine-tuned neurons $\mathbf{y}_{\text{m}}$ in the neural network that tune their receptive fields to more relevant sensory objects $\mathbf{x}_{\text{multi}}$ that are also sensed from multimotor concepts of the event.*

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

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 $\mathbf{z}_{\text{innate}}$ be replaced by $\mathbf{z}_{\text{multi}}$ and $\mathbf{x}_{\text{effect}}$ by $\mathbf{x}_{\text{multi}}$. 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) by $\mathbf{y}_{\text{m}}$, we have the above expression. Note: the machine continues to run DN. □

Theorem 4.1 can be verbally summarized as "practice makes perfect". For example, to learn how to drive cars one must try driving. This process also includes book reading.

Condition to start imitation. The condition in our above discussion is sensory $\mathbf{x}_{\text{multi}}$, which can represent also time. Time is a concept that the learner can learn through imitation and counting. The speed of context change of "where-and-what" of each 3D object, represented as last "where-and-what" in $\mathbf{z}_{t-1}$, the internal context $\mathbf{y}_{t-1}$, and the current sensory vector $\mathbf{x}_{t-1}$. The next context $\mathbf{z}_t$ represents the change, or motion, of the attended 3D object. Thus, the concept of motion or time emerges as the condition to start imitation.

The starting time of imitation could be triggered by an environmental cue, e.g., teacher's nod or the setting for doing homework is ready (e.g., clock).

Imitation of internal attention. Imitation for internal attention is not motor-imposable, since there is no overt motor that corresponds to the attention. Suppose the teacher demonstrates "notice pedestrians". However, it is impractical for the teacher to motor-impose "pedestrian" since vocal tract is inside the body, not imposable. However, if the learner has spoken "pedestrian" while its attention is on pedestrian, teacher's demonstration "pedestrian" (e.g., speak) causes "pedestrian" motor in the learner to fire. The firing boosts internal Y neurons through top-down connections so that the learner attends to pedestrians. Let us call this type of internal imitation language-directed imitation of internal attention.

Generality and creativity of imitation. An imitation of 3D events involves attending to a few concepts and their relationships but exchanging some associated concepts, all of which have been learned by the conscious learner represented in its motor area.

In Fig. 1, three concepts are attended to: hand, cup, and mouth, and two concept-relationships are attended to, cup-in-hand and cup-at-mouth. Two concepts are associated as human type but substituted, "I" substitutes "teacher".

Using the generalization reasoning above, we can see that imitations are generally applicable to any observed events.

Theorem 4.3 (Generality 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 2.1 involves learning a universal Turing machine modeled as context transitions in Eq. (3). According to Theorem 4.2, 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 $\mathbf{x}_{\text{multi}}$ and then to create a program as a sequence of motor signals in the form of $\mathbf{z}_{\text{multi}}$, and finally to carry out the program back to the real world. Such compositions of programs correspond to human thoughts [35]. Therefore, the context transitions in Theorem 4.2 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. The theory of creativity is for any subject matters.

Albert Einstein's work on general relativity is a result of autonomous imitations through trial and error. Similarly, all physical experiments that verified the relativity theory are also autonomous imitations. However, the former is harder because the gap from the 3D events (published physics experiments that Albert Einstein had learned then) to the created program (the relativity theory paper) is significantly wider than the latter (from the published relativity paper to a plan to verify the correctness). Therefore, research awards should not be based on primarily how complete an experiment is, but on also how large the gap is between prior arts and a novel result and its impact.

5 DISCUSSION

This work 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 [25] is that the learner observes teacher's demonstrations using its sensors without motor-imposed training. Can such imitations reach a school level?

Experiments for conscious learning require brain-scale on-the-fly learning hardware. This line proposes a future of electronics engineering—brainoid chips for on-the-fly learning.

Using this fundamentally new method, artificially machines will be able to automatically acquire human-like consciousness, from machine infancy to adulthood, with helps of human teacher demonstrations or without, since a real world also demonstrates its facts. However, although [23] proved that this process by DN is optimal in the sense of maximum likelihood, rediscovery of human knowledge, such as Newtonian physics and Einstein's general relativity by machines requires very long time and much resource, but is now possible in theory. Therefore, for all practical purposes, humans will establish robot schools to teach robots through demonstrations, e.g., robot driving schools. However, the current lack of brain-scale real-time learning chips and the lack of such AI education are two bottlenecks.

REFERENCES

[1] M. Cole and S. R. Cole. 1996. *The Development of Children* (3rd ed.). Freeman, New York.

[2] M. B. Feller, D. P. Wellis, D. Stellwagen, F. S. Werblin, and C. J. Shatz. 1996. Requirement for cholinergic synaptic transmission in the propagation of spontaneous retinal waves. *Science* 272, 5265 (1996), 1182–1187.

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

[4] A. Graves, G. Wayne, and I. Danihelka. 2014. *Neural Turing Machines.* Technical Report arXiv:1410.5401. Google DeepMind, London, UK.

[5] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwinska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, A. P. Badia, K. M. Hermann, Y. Zwols, G. Ostrovski, A. C., H. King, C. Summerfield, P. Blunsom, K.

Kavukcuoglu, and D. Hassabis. 2016. Hybrid computing using a neural network with dynamic external memory. *Nature* 538 (2016), 471–476.

[6] E. H. Hess. 1973. *Imprinting: Early Experience and the Developmental Psychobiology of Attachment.* Van Nostrand Reinhold Company, New York.

[7] S. Hochreiter and J. Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[8] J. E. Hopcroft, R. Motwani, and J. D. Ullman. 2006. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, Boston, MA.

[9] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne. 2017. Imitation Learning: A Survey of Learning Methods. *Comput. Surveys* 50 (June 2017), 1–35.

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

[11] Y. LeCun, L. Bengio, and G. Hinton. 2015. Deep Learning. *Nature* 521 (2015), 436–444.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-Based Learning Applied to Document Recognition. *Proceedings of IEEE* 86, 11 (1998), 2278–2324.

[13] J. C. Martin. 2003. *Introduction to Languages and the Theory of Computation* (3rd ed.). McGraw Hill, Boston, MA.

[14] A. N. Meltzoff and M. K. Moore. 1977. Imitation of Facial and Manual Gestures by Human Neonates. *Science* 198, 4312 (Oct. 7 1977), 74–78.

[15] J. Moran and R. Desimone. 1985. Selective attention gates visual processing in the extrastrate cortex. *Science* 229, 4715 (1985), 782–784.

[16] B. A. Olshausen, C. H. Anderson, and D. C. Van Essen. 1993. A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *Journal of Neuroscience* 13, 11 (1993), 4700–4719.

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

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115 (2015), 211–252.

[19] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. 2007. Robust object recognition with cortex-like mechanisms. *IEEE Trans. Pattern Analysis and Machine Intelligence* 29, 3 (2007), 411–426.

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

[21] M. Tomasello. 2003. *Constructing a Language: A Usage-Based Theory of Language Acquisition.* Harvard University Press, Cambridge, MA.

[22] A. M. Turing. 1936. On computable numbers with an application to the Entscheidungsproblem. *Proc. London Math. Soc., 2nd series* 42 (1936), 230–265. A correction, *ibid.*, 43, 544-546..

[23] J. Weng. 2015. Brain as an Emergent Finite Automaton: A Theory and Three Theorems. *International Journal of Intelligence Science* 5, 2 (2015), 112–131.

[24] J. Weng. 2020. Autonomous Programming for General Purposes: Theory. *International Journal of Huamnoid Robotics* 17, 4 (August 2020), 1–36.

[25] J. Weng. 2020. Conscious Intelligence Requires Developmental Autonomous Programming For General Purposes. In *Proc. IEEE Int. Conf. on Dev. Learning and Epigenetic Robotics.* Valparaiso, Chile, 1–7.

[26] J. Weng. 2021. On Post Selections Using Test Sets (PSUTS) in AI. In *Proc. International Joint Conference on Neural Networks.* Shengzhen, China, 1–8.

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

[28] J. Weng and M. Luciw. 2009. Dually Optimal Neuronal Layers: Lobe Component Analysis. *IEEE Trans. Autonomous Mental Development* 1, 1 (2009), 68–85.

[29] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen. 2001. Autonomous Mental Development by Robots and Animals. *Science* 291, 5504 (2001), 599–600.

[30] J. Weng, Z. Zheng, and X. Wu. 2019. Developmental Network Two, Its Optimality, and Emergent Turing Machines. U.S. Patent Application Number: 16265212. Approval pending.

[31] J. Weng, Z. Zheng, X. Wu, J. Castro-Garcia, S. Zhu, Q. Guo, and X. Wu. 2018. Emergent Turing Machines and Operating Systems for Brain-Like Auto-Programming for General Purposes. In *Proc. AAAI 2018 Fall Symposium: Gathering for AI and Natural Systems.* Arlington, Virginia, 1–7.

[32] P. J. Werbos. 1994. *The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting.* Wiley, Chichester.

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

[34] X. Wu and J. Weng. 2020. Muscle Vectors as Temporally "Dense Labels". In *Proc. International Joint Conference on Neural Networks.* Glasgow, UK, 1–8.

[35] X. Wu and J. Weng. 2021. On Machine Thinking. In *Proc. International Joint Conf. Neural Networks.* IEEE Press, Shenzhen, China, 1–8.