Brain-Like Emergent Temporal Processing: Emergent Open States

Juyang Weng *Fellow, IEEE*, Matthew D. Luciw *Member, IEEE*, and Qi Zhang

***Abstract*—Informed by brain anatomical studies, we present the Developmental Network (DN) theory on brain-like temporal information processing. The states of the brain are at its effector end, emergent and open. A Finite Automaton (FA) is considered an external symbolic model of brain's temporal behaviors but the FA uses handcrafted states and is without "internal" representations. The term "internal" means inside the network "skull". Using action-based state equivalence and the emergent state representations, the time driven processing of DN performs state-based abstraction and state-based skill transfer. Each state of DN, as a set of actions, is openly observable by the external environment (including teachers). Thus, the external environment can teach the state at every frame time. Through incremental learning and autonomous practice, the DN lumps (abstracts) infinitely many temporal context sequences into a single equivalent state. Using this state equivalence, a skill learned under one sequence is automatically transferred to other infinitely many state-equivalent sequences in the future without the need for explicit learning. Two experiments are shown as examples: The experiments for video processing showed almost perfect recognition rates in disjoint tests. The experiment for text language, using corpora from the Wall Street Journal, treated semantics and syntax in a unified interactive way.**

***Index Terms*—Brain-mind architecture, representation, attention, transfer, perception, cognition, behavior, computer vision, text understanding, time warping, sequential abstraction, regression, complexity.**

I. Introduction

THE artificial intelligence community has largely followed a path of handcrafted symbolic representation: Given a task to be executed by a machine, it is the human designer who understands the task and hand picks the concepts as symbols required by the task before any machine learning can take place.

The wave of artificial neural network (ANN) research in the 1980's represented the rise of the connectionist approach [82], [63]. Using an ANN, the internal representation of the network can emerge through incremental learning. The human designer specifies the formats for the network input ports and the output ports as well as the internal learning mechanisms, but not the actual internal representations. The internal representations generated by some ANN are distributed, instead of symbolic. By "distributed" we mean that the presence of an extra-body concept (e.g., "car") does not have an iff (if and only if) condition with the firing of any single internal element (e.g., neuron or module), since such an iff condition requires handcrafting. Note: an iff condition is not the same as one-to-one mapping, since many other neurons are also allowed to fire when the concept is present. However, the term "connectionist model" is misleading since it does not effectively distinguish itself from symbolic networks (SNs), especially probabilistic versions of SNs. In a probabilistic SN (e.g., HMM), a variable about an extra-body concept is also distributed over multiple nodes. Therefore, we use the term "emergent networks" (ENs) here, which is stricter than ANNs and connectionist approaches.

Juyang Weng is with Michigan State University, East Lansing, MI, USA (weng@cse.msu.edu) and holds a Changjiang visiting professor position at Fudan University, Shanghai, China. Matthew D. Luciw is with Michigan State University (luciwmat@cse.msu.edu). Qi Zhang is with the School of Computer Science, Fudan University, Shanghai, China (email: qi_zhang@fudan.edu.cn). JW conceptualized and drafted the paper. ML did the experiments in Sec. VIII.A. QZ did the experiments in Sec. VIII.B.

A. Autonomous mental development

From the viewpoint of biological development [19], [76], [103], the representation inside the "brain" is *epigenetically* generated throughout the lifetime of each individual life. "Epi" here means "after". By "epigenetic", we mean that the genes inside every cell regulate, but do not totally determine, the developmental process of the body and the brain through which the body and the brain interact with the internal and external environments after the conception.

This implies that the representations inside the brain for extra-body concepts fully autonomously emerge inside the brain's closed skull. Humans teachers, as part of the external environment (i.e., outside the brain including the body), are able to interact with the brain through only its sensory ports and effector ports. Weng 2012 [99] argued that all representations fall into two categories, symbolic and emergent. Symbolic representations, as defined later, are task specific, since it is requires that a task be given and it is the human designer who understands the task and who handpicks a static set of symbolic task concepts about the extra-body task environments. Neural network seems the only known way in which internal representations can autonomously emerge. Therefore, neural networks are necessary for autonomous mental development (AMD), not optional. We will discuss below a new definition for emergent representation, which is stricter than ANN and connectionist approaches. Those ANNs that allow a human designer to impose symbolic extra-body meanings into part of networks (e.g., handcrafted features such as SIFT features and Gabor features) belong to symbolic models by the new definition.

This line of reasoning further implies that the *Developmental Program* (DP) — the functions of the biological genome — for an animal is task nonspecific, as argued by Weng et al. 2000 [103], because "task" is largely an extra-body concept. The genome seems to reliably regulate the generation of infra-body behaviors (e.g., inborn sucking behavior from a touch on baby's lip) but does not seem to rigidly determine the representations for extra-body concepts (e.g., not for extra-body concept "nipple" since sucking behavior also responds to a touch by a stick on baby's lip). In particular, the DP

should not have a handcrafted model about the extra-body world. The actual extra-body environment of a species at any time is highly dynamic, highly unpredictable, and open-ended. As argued in Weng 2012 [99], we hypothesize that for an emergent model about the brain, the DP may embed inborn behaviors for intra-body concepts (e.g., sucking on a touch on the lip) but does not need to model any extra-body concepts (e.g., oriented edge of an extra-body object). All representations for extra-body concepts emerge through interactions with the extra-body environment. Such extra-body concepts include type, location, scale, owner, retail price, and task goal.

Whether a developmental agent can successfully acquire a mental skill (or knowledge) depends on 5 factors: (1) the sensors, (2) the effectors, (3) the DP, (4) the computational resource, and (5) the experience.

"Effector" is a general term that includes muscles and glands. The term "motor" is often used to replace effector, although its meaning is often restricted to motor effectors (muscles), not necessarily including glands.

Regulated by a DP, the developmental agent has its "skull closed" throughout the lifetime. The "skull" of the network encapsulates the network from its external physical environment, leaving its sensory ends and its motor ends open to the external environment (other than the brain). Note that the body of the agent is also included in this external environment, since it is outside the "skull". A developmental agent must incrementally develop mental skills for an open variety of tasks without requiring a human to re-program the DP or to directly manipulate inside the brain after the agent "birth".

B. Prior temporal models

Marvin Minsky 1991 [65] and others argued that symbolic models are logical and neat, but connectionist models are analogical and scruffy. At the David Rumelhart Memorial talk August 3, 2011 during the International Joint Conference on Neural Networks, Michael Jordan correctly stated that prior neural networks do not abstract well and symbolic models had been better than prior neural networks in terms of abstraction.

With regard to *temporal information* processing, Buonomano & Merzenich [9] proposed a randomly connected network that translates temporal information (i.e., sound) into its spatial representation. The biologically recorded neuronal learning phenomenon of Spike Timing-Dependent Plasticity (STDP) [5], [14] spans a time interval of 50ms. However, this short time span is not sufficient, as correctly pointed out by Drew & Abbot [17], to explain how the brain deals with longer temporal dependency. Mauk & Buonomano [62] argued that the brain uses its intrinsic mechanisms to deal with time, and it does not have explicit delay lines and does not have a global clock. Drew & Abbott [17] proposed that the gradual change in the level of membrane potential inside a neuron may record some temporal information. However, this seems also not sufficient and robust for long time dependency as pointed out by Ito et al. 2008 [38]. How the brain deals with long time context is still elusive, especially considerably beyond around 30 ms modeled by STDP.

Neuro-anatomic studies [24], [10], [11] have demonstrated that the brain is not a cascade of brain areas, but a complex network of areas. Between any two connected areas, the connections are universally bi-directional, i.e., two unidirectional bundles, with few exceptions (e.g., there is no top-down connections from LGN to the retina in primates but not so with other animals). Using such properties, the Multilayer In-Place Learning Networks (MILN) [101], [102], [55] and the general-purpose visual processing networks Where What Networks (WWN) [45] have shown their properties of abstraction and attention through the assistance of their motor signals. This indicates that such a brain-inspired model seems to be a general developmental model for processing *spatial information*.

The exiting SNs and ENs for temporal processing suffer from some major problems, such as the long-term memory loss problem, the lack of power for abstraction, and the lack of power for transfer, as we will discuss later in Section IV.

C. Novelty and importance

This paper presents the Temporal Context Machine (TCM), an experimental embodiment of a general brain-mind model called Developmental Networks (DN) [94], [97], [98] which deals with both space and time information. The embodiments of DN include WWN-1 [45], WWN-2 [44], WWN-3 [56], WWN-4 [57], WWN-5 [86], and TCM [105]. TCM is also an extension to temporal domain from the spatial networks MILNs [102]. This is an archival theoretical paper, growing out of the theoretical work of Weng 2010 [95], citing as two examples the visual experimental studies originally published in Luciw & Weng 2008 [58] and the language experimental studies first appeared in Weng et al. 2009 [105].

This work is a departure from (1) existing models in traditional artificial intelligence that use handcrafted symbolic representations (e.g., SNs) and (2) existing connectionist models that use emergent representations (e.g., ENs). Compared with those models, the major novelty of the work includes the following aspects.

Each state of TCM is open as input and output: By default, a state here means primary state, at the motor end of the TCM, open to the external world. The state of an internal area is called internal state, but not primary. Based on neuroanatomy (e.g., [24]) and cortical recordings (e.g., [6]), we hypothesize that a major purpose of the brain is to generate actions that are open to the external world (i.e., outside the skull) — observable, teachable and calibratable. Internal states in an adult brain serve the need of the states at the motor end. A major origin of the internal representations are the external environment, via the sensory end and the motor end. The sensory end is largely "supervised" by the external environment, although only a part of it is attended at any time. Therefore, the states of TCM are at the effector end which are open to the external world, so that the external world (e.g., teachers) can observe, supervise, and calibrate the state at each frame time whenever such an interaction is practical. Namely, the effector ports are not only for outputs, but also for inputs. This is in contrast with existing connectionist temporal models, such as the Jordan Network [46], the Elman Network

[18], the Long Short-Term Memory (LSTM) [35], the Liquid State Machines (LSM) [61], the Echo State Networks (ESN) [41], Continuous Time Recurrent Neural Networks (CTRNN) [109], and the Reservoir Computing [60], whose recurrence takes place inside an internal hidden area and, thus, their states are hidden from the external world, not directly observable and interactively teachable by the external world. This results in their lack of effective mechanism for state equivalence and skill transfer (discussed below). The theory here also includes hidden internal states (i.e., the response of the hidden area Y of TCM), but open states as actions are primary and physically causal for the brain. In contrast, as we will see below, hidden states are secondary and assistive to the open, primary states. In our following discussion, states of TCM mean the primary, open state by default.

TCM states are rich in meanings: All brain skills are eventually expressed as the states of effectors — muscles and glands. Psychologists argued that brain skills can be divided into two categories, explicit (or declarative) and implicit (or manipulatory) [80], [88]. Explicit skills can be expressed in a natural language — written, said, or signed (e.g., the American Sign Language). Implicit skills do not have a natural language but their effects are delivered through manipulations (e.g., run, grasp, and dance). Therefore, the effector ports of the brain are hubs for meanings, as far as the entire brain is concerned. The meanings include, but not limited to, goal, intent, value, spatial context, temporal context, and actions. Each TCM state is not limited to a single meaning, but represents multiple concurrent meanings of an attended object (e.g., plant, fruit, apple, red, at upper-left location). We hypothesize that major human brain skills can be expressed through the states of effector ports (i.e., muscles and glands) since they are the major outputs of the brain (other than negligible outputs such as EEG and heat).

TCM emulates Agent Finite Automata: Traditionally, various automata as language acceptors [36] have been used for checking syntax, not semantics. This work extends the framework of deterministic finite automaton DFA (or FA for short) to agent FA (AFA), which outputs its symbolic state instead of yes or no for sentence acceptance. Each symbol, either an input symbol or an output state, has a set of associated meanings expressed by a natural language in the design document. The AFA enables us to model and understand how a TCM abstracts and reasons. Simulating any AFA, the representation in the motor area of TCM as a state is recursively used as temporal top-down context for subsequent processing.

Network sequential abstraction: Motor outputs can be abstract, where "abstract" means that each output depends on only related spatiotemporal context of the input sequence. Each TCM has a fully emergent internal representation. The autonomy in the emergence of internal representation in neural networks has posed a great challenge for network abstraction. However, unlike prior connectionist models criticized by Minsky and Jordan, in principle a TCM can abstract as well as any other Symbolic Networks (SN). By SN, we mean a symbolic model that uses states as equivalent classes for temporal contexts. As reviewed in Sec. III-B, many practical symbolic models are SNs. The new work here shows that a TCM can emulate any complex AFA. In other words, a TCM can abstract as well as any SN. This seems the first theoretical work that proves that an emergent network (TCM) can perform abstraction at least as powerful as SNs. Such a departure from symbolic models seems to be also useful for understanding how the brain-mind abstracts through time.

Skill transfer: The emergent TCM can perform transfer: transfer of a learned skill to many other settings without a need of explicit learning. The state equivalence in the AFA theory is the basis of transfer. Suppose that a TCM takes an input word (e.g., "cat"), it generates an action (e.g., report "kitten"). This is called a skill. However, such a skill is not necessarily applicable to all context sequences. In our example, the skill of reporting "kitten" upon input "cat" is applicable to context "young". A TCM forms equivalent spatiotemporal (numeric vector) states and learns *the skills* conditioned on each state, so that one skill learned from a particular context sequence can be correctly transfer to infinitely many equivalent context sequences in the future without a need for explicit learning.

TCM Properties: The new work here proves a series of properties of TCM, including AFA simulation, context dependent attention, active time warping, temporal attention, time duration, skill transfer, and complexity.

Unified spatial and temporal processing. With this unification, this work makes the following prediction about the brain network:

> *The spatial brain network seems not only a general-purpose engine for developing rich capabilities for* spatial information processing *but also a general-purpose engine for developing rich capabilities for* temporal information processing*, without any components exclusively dedicated to time.*

This prediction requires much future multi-disciplinary work to fully verify.

For intuition in our discussion, we consider text as an example of the sensory modality for our discussion. However, in principle, this brain-inspired model is not limited to the text modality as it is applicable to any sensory modality. We will also discuss our corresponding visual processing studies below. In contrast with traditional text processing methods which treat each text input as a static symbolic string for batch processing, the TCM discussed here scans its sensory input port in time, as a binary image, like a brain's eye. In the text experiments, we assume that the sensory input port receives a word at a time. This mode is suitable here before fully autonomous robotic camera saccades become a reality in the future, e.g., when a robot is mature enough to autonomously read a book using its pan-tilt head.

We will mainly use motor-supervised learning to explain how a TCM learns. For example, while a teacher holds a child's hand to teach drawing, the child is conducting motor-supervised learning. Of course, motor-supervised learning is tedious, but this mode must be realized by the brain-mind before it can self-practice (self-generated motor teaching) and use its motivational system (e.g., the dopamine and serotonin systems modeled in [73], [13]). It seems also wrong to think that a human agent is a completely autonomous agent, since other humans supervise his motors (e.g., when shake hands)

and his perception and cognition (e.g., parent teaching and school education). A full autonomy is the self-organization inside the brain (inside the skull), not outside the brain.

The remainder of this paper is organized as follows: First the TCM algorithm is presented in Section II. Section III discusses different types of representation to motivate the emergent representation used by TCM. Section IV discusses different temporal mechanisms. The TCM is further analyzed in Section V. Several experimental results with video and text data sets are presented in Section VIII to indicate the performance. Finally, Section IX provides some concluding remarks.

II. Algorithm

This algorithm describes a brain-like network TCM that consists of three areas, X, Y, and Z, as illustrated in Fig. 1, where the connection patterns are also shown.

The area X is Y's sensory area. For image input of $m \times n$ pixels, X has $d = m \times n$ neurons. For text input, each word is mapped to a different input pattern as an image. In general, X can be any area that are closer to sensory inputs than Y. For example, if Y is V2, then X can be LGN and V1 combined.

The area Z is Y's motor area. The number of neurons in Z is the number of "muscles" elements or muxels for short. If Y is the premotor area, then Z is the primary motor area. Using a one-to-one symbol-vector mapping, the number of neurons in Z is the limited resource in Z. In general, multiple neurons in Z can fire to represent more firing patterns. In our experiments, the human teacher in the external environment let each muscle neuron in Z represent one concept value. This corresponds to a simple language (e.g., raising the i-th figure to represent the i-th meaning).

The area Y is the bridge for its two banks X and Z. The major function of the bridge is to predict the signals in its two banks. Y provides features in $X \times Z$ and reports top matches. The more neurons an area Y has, the more finely the area can tessellate the manifold of $X \times Z$ in which the observed samples lie.

A biological network grows and operates (develops in general) in the following way. Neurons in the brain fall into two broad categories [47, p. 329], excitatory projection neurons and inhibitory inter-neurons. Projection (i.e., feature) neurons grow their dendrites and axons to near and far locations. Within each area, there are many inter-neurons which can only connect locally. Inhibitory inter-neurons turn their connected pre-synaptic axon into its own "opposite" output: The neural transmitters from an inhibitory pre-synaptic inter-neuron inhibit the firing of the post-synaptic neuron. Effectively, inhibitory interneurons make projection neurons to inhibit mutually so that at any time, in each area only few projection neurons whose pre-response potentials are top can survive the competition and fire. To avoid time-consuming back-and-forth inhibitions through inter-neurons, we use a top-k competition mechanism which quickly sorts out the top winner projection neurons. Using the top-k competition in each area, by neurons in the following, we mean excitatory projection neurons.

Each pyramidal neuron autonomously determines from which pre-synaptic neurons to connect, based on its Hebbian learning mechanism. Since the weights of each neuron are random originally, its firing is also random to start with. As we can see later mathematically, the Hebbian learning mechanism of each neuron allows only synapses with those pre-synaptic neurons that often co-fire with the post-synaptic neuron to survive and all other synapses die out. The connection patterns in Fig. 1 result after sufficient interactions with the external environment after the birth.

We predict that this 3-area network is a simplified "brain", where the brain is represented by the area Y. We hypothesize that the complex brain structure emerges through interactions between the internal and external environments.

A. TCM Algorithm

Each TCM has three areas, X, Y and Z. Y is always hidden. X is exposed to the external environment as it connects with sensors. Likewise, Z is exposed as it is connected with effectors. For each area A in $\{X, Y, Z\}$, let $N = (V, G)$ denote the adaptive part of area A, where $V = (\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_c)$ contains the weight vectors of the c neurons in area A and $G = (n_1, n_2, ..., n_c)$ contains the firing ages of the neurons (the number of times a neuron has fired). The firing age n_j of a neuron j is the number of times the neuron has fired.

Algorithm 1 (TCM): Let f_Y indicate the area function of every area of TCM.

1) At time $t = 0$, for each area A in $\{X, Y, Z\}$, initialize its adaptive part $N = (V, G)$ and the response vector $\mathbf{r}$, where V is the synaptic weights and G the neuronal ages.
2) At time $t = 1, 2, ...$, for each area A in $\{X, Y, Z\}$, do the following two steps repeatedly forever:
   a) Every area A computes using area function f.
   $$(\mathbf{r}', N') = f(\mathbf{b}, \mathbf{t}, N) \quad (1)$$
   where f is the unified area function; $\mathbf{b}$ and $\mathbf{t}$ are area's bottom-up and top-down inputs, respectively, from the current network response; and $\mathbf{r}'$ is the new response of area A.
   b) For each area A in $\{X, Y, Z\}$, A replaces: $N \leftarrow N'$ and $\mathbf{r} \leftarrow \mathbf{r}'$.

Since we consider that the area X is supervised directly by the external environment, we do not need to compute the area function for X. However, if Y is a Brodmann area inside the brain, X and Z are its two input areas. Then, X computes so that Y predicts the signals in X. Likewise, some or all components of Z can be supervised by the external environment, typically as the temporal abstract context at that time t. Therefore, if Y is the entire brain, X takes in a sensory movie and Z outputs and inputs a motor movie, as illustrated in Fig. 1.

B. The area function

Next, we describe the area function f in Eq. (1). We use $\dot{\mathbf{v}}$ to denote the unit vector of a vector $\mathbf{v}$. $\dot{\mathbf{v}} = \mathbf{v}/\|\mathbf{v}\|$.

Each neuron in area A has a weight vector $\mathbf{v} = (\mathbf{v}_b, \mathbf{v}_t)$, corresponding to the area input $(\mathbf{b}, \mathbf{t})$, if both bottom-up part

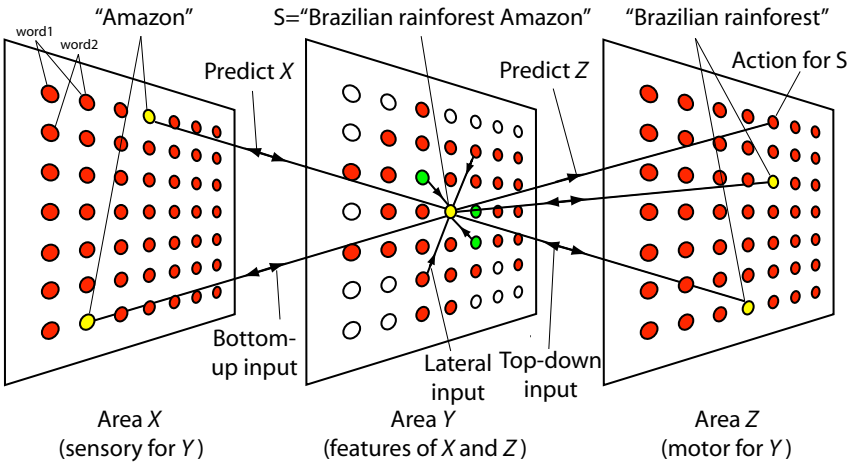


Fig. 1. The basic architecture of TCM. In this illustration, text as an image for text processing, but the area X can contain any sensory modality in principle — visual, auditory, somatosensory, olfactory, and taste. The area Z can contain any effector modality in principle — muscles and glands. Only input connections to the center neuron in area Y are shown, but all other neurons in X, Y and Z are connected in a similar way. Area X gives the current bottom-up input (e.g., word as an image) $\mathbf{x}(t)$ to area Y. In general, $\mathbf{x} \in X$ is an image with many neurons (pixels) firing. Area Z gives the previous response or externally supervised image $\mathbf{z}(t)$ as the spatiotemporal state (context). Area Y takes input $(\mathbf{x}(t), \mathbf{z}(t))$ to generate response as $\mathbf{y}(t+1)$. Within area Y, lateral connections are present, excitatory (green), inhibitory (red), and none (white). Areas X and Z compute in a similar way. If Y is the entire brain, X does not have bottom-up source and Z does not have top-down source. Each bi-directional arrow represents two connections in opposite directions.

and top-down part are applicable to the area. Otherwise, the missing part of the two should be dropped from the notation. Its pre-response value is the sum of two normalized inner products:

$$r(\mathbf{v}_b, \mathbf{b}, \mathbf{v}_t, \mathbf{t}) = \dot{\mathbf{v}} \cdot \dot{\mathbf{p}}, \tag{2}$$

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

It is important that the area A only updates the best-matched memory and keeps all memories as the long-term memory for this input $\mathbf{p}$. To do this, only top k winners fire and update. To avoid slow iterative updates in sorting out the winners in A, we use explicit sort in our simulation. Considering $k = 1$ (necessary when the memory is small), the winner neuron j is identified by:

$$j = \arg \max_{1 \le i \le c} r(\mathbf{v}_{bi}, \mathbf{b}, \mathbf{v}_{ti}, \mathbf{t}). \tag{3}$$

The winner fires with $r_j = 1$ (a spike). All other neurons in A do not fire, $r_i = 0$ all $i \neq j$. In an ideal case later, the top-1 match is always perfect: $\dot{\mathbf{v}}_j = \dot{\mathbf{p}}$. The winner-take-all corresponds to the situation that all lateral connections are inhibitory (i.e., all other neurons in the same area are red in Fig. 1).

C. Learning of the area function

TCM learns while performing. That is, the learning is incremental, online, in real time, and on the fly. It is impractical for the brain, natural or artificial, to store sensory frames as a batch before learning takes place.

All the synapses of each neuron learn incrementally based on Hebbian learning — cofiring of the pre-synaptic activity $\dot{\mathbf{p}}$ and the post-synaptic activity r of the firing neuron. The synaptic vector of a firing neuron has a gain $r\dot{\mathbf{p}}$. Other non-firing neurons do not modify their memory. When a neuron j fires, its weight is updated by a Hebbian-like mechanism:

$$\mathbf{v}_j \leftarrow w_1(n_j)\mathbf{v}_j + w_2(n_j) r_j \dot{\mathbf{p}} \tag{4}$$

where $w_2(n_j)$ is the learning rate depending on the firing age n_j of the neuron j [100] and $w_1(n_j)$ is the retention rate with $w_1(n_j) + w_2(n_j) \equiv 1$.

The simplest version of $w_2(n_j)$ is $w_2(n_j) = 1/n_j$, which has a "zero" momentum even when $\mathbf{v}_j$ is initialized by a random vector which is what we did in the experiments. When a neuron learns for the first time, $n_j = 1$, $w_2(n_j) = 1$ so that $\mathbf{v}_j = 0 + 1 \cdot 1 \cdot \dot{\mathbf{p}} = \dot{\mathbf{p}}$, immediately memorizing the input vector $\dot{\mathbf{p}}$ so that the neuron must be the winner in the next practice, perfectly responds to the $\dot{\mathbf{p}}$. This fast learning mechanism contributed to the surprisingly fast learning results in experiments, reported in Section VIII, when the number of neurons in TCMs is significantly smaller (about 60% short) than the number of Y neurons needed to perfectly memorize all the $\dot{\mathbf{p}}$ vectors.

The age of the winner neuron j is incremented $n_j \leftarrow n_j + 1$. All other neurons in the area do not respond and do not advance their firing ages.

D. Explanation of the area function

From Eqs.(2) and (3), we can see that if the number of neurons is sufficiently large, the nearest neighbor $\mathbf{v}_j = (\mathbf{v}_{bj}, \mathbf{v}_{tj})$ typically matches each unobserved input $\dot{\mathbf{p}} = (\dot{\mathbf{b}}, \dot{\mathbf{t}})$ well

$$\dot{\mathbf{v}}_{bj} \approx \dot{\mathbf{b}} \text{ and } \dot{\mathbf{v}}_{tj} \approx \dot{\mathbf{t}} \tag{5}$$

and each observed input perfectly. In other words, for a neuron to fire, the bottom-up vector and top-down vector must both match well. As we will see, using three areas X, Y, and Z each of which computes in the above way, the TCM becomes a content addressable memory for bottom-up input $\mathbf{x}$ and top-down input $\mathbf{z}$. Furthermore, a partial $\mathbf{x}$ or a partial $\mathbf{z}$ as input will enable other parts of $\mathbf{x}$ and $\mathbf{z}$ to pop up with the corresponding recall, where "partial" means many components are left "free".

However, the above conclusion also depends on how the c neurons in area A distribute, which is the subject of the following subsection.

E. Optimality of the area function

The theory behind the area function is the Lobe Component Analysis (LCA), which is a dually optimal model for updating a neuronal area A using Hebbian learning. LCA is similar to SOM in what it does, but LCA is optimal. It incrementally learns the weights using the best direction of change and the best amount of change to minimize the expected error in representing the incoming high dimensional inputs of the area in relation to its previous experience. In doing so, it optimally deals with the two conflicting needs during incrementally learning: the need to immediately learn the current area input $\dot{\mathbf{p}}$ by altering the limited memory in the area A and the need for area A to keep the long-term memory stable for past experience.

As analyzed and proved in [100], the c vectors in V move, in the best incremental way under limited training experience, to the best target in the lower dimensional manifold of the input space P of $\dot{\mathbf{p}}$ in which the observed samples lie. In particular, the regions in P from which no samples arise do not waste neurons.

The rate profile $w_2(n_j) = 1/n_j$ means that $\mathbf{v}_j$ is the recursive mean of input $\dot{\mathbf{p}}$:

$$\mathbf{v}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} \dot{\mathbf{p}}(t_i) \tag{6}$$

where t_i is the firing time of the neuron. A component in the gain vector $r_j \dot{\mathbf{p}}$ is zero if the corresponding component in $\dot{\mathbf{p}}$ is zero. Thus, all those potential connections in which the presynaptic neuron has never co-fired with the post-synaptic neuron do not really exist, resulting in the typical sparse connections illustrated in Fig. 1. Incrementally computed this way, each component in $\mathbf{v}_j$ is the estimated probability for the pre-synaptic neuron to fire under the condition that the post-synaptic neuron fires.

Supported by the above property, the area function has the following two optimality properties proved by [100]:

1) Spatial optimality: The theoretical target of the set V^* of synaptic vectors of area A is the best in minimizing the representation error of the area input space P using a limited number of c neurons. For area Y, $P = X \times Z$, the parallel input space of ascending input space X and the descending input space Z.
2) Temporal optimality: The update direction and step size for the winner neuron j are best to minimize the expected distance between the estimated $V(t)$ at time t and its theoretically best, but unknown, target V^*.

The spatial optimality implies that each area A as the bridge has the smallest possible expected error in representing its two banks. The temporal optimality means that the area A learns fastest using a limited amount of learning experience up to time t. A full review of the LCA theory, e.g., firing age dependent Hebbian learning, is beyond the scope of this paper. The reader is referred to Weng & Luciw 2009 [100].

This seemingly simple TCM algorithm is very challenging to understand, having rich properties, and is of general purpose in dealing with time. In the remainder of this paper, we discuss these subjects.

III. Representations

We first discuss some basic concepts that are closely related to representations.

A. Basic concepts

1) Embodiment: The term *embodiment* means having a body. A brain in a vat [8] is disembodied. Sensors and effectors are important body organs for an embodied brain. Much of the remaining body organs provide the energy and service needed for the normal operation of the brain and the body. The work here addresses how an embodied brain, natural or artificial, can interact with its external environment via its sensors and effectors on its body. The scope of the work here includes theory, algorithm, and robot simulations, but not including experimental tests on a real robot.

2) Grounding: Another concept is *grounding*, which means that the sensors and effectors must interact directly with the actual external task environment — the real physical environment — without a human in between. However, the real physical world does not provide a means for precise timing and performance evaluation. Thus, realistic robotic simulations are necessary for our examples of performance evaluation.

3) Discrete time: A grounded life, biological or artificial, lives in the real physical environment in real time. Emulated by a digital computer, we require that the brain is sampled at discrete times $t = t_0, t_1, ..., t_n$, where t_0 is the conception time, t_n is the death time, $t_{i+1} = t_i + \Delta$, $i = 1, 2, ...n-1$, and Δ is a constant. For notation simplicity, we write $t = 0, 1, ..., n$.

4) Discrete time vs. continuous time: It is important to know that when Δ is small enough (e.g., $\Delta = 1$ms), a discrete brain model can model the brain at a desired temporal precision. Although not using all infinitely many time instances between two consecutive time samples, the discrete-time models here seem to be more mathematically accurate than the continuous-time counterparts, since it does not use derivatives to approximate differences. For example, the discrete-time approach in TCM is more accurate than the discrete-time approach in LSM, at least in terms of the applicability of the formulation to practice.

5) The sensory-movie-and-motor-movie loop: Through the life, the agent acquires a sensory movie as $\{\mathbf{x}(t), t = 0, 1, ...\}$ and produces a motor movie as $\{\mathbf{z}(t), t = 0, 1, ...\}$. But these two movies are grounded and embodied in the real physical world, highly dependent on each other and highly dependent on the external environment. For developmental reasons above, the DN should not use a handcrafted model about the world, but the sensory movie is greatly affected by the motor-movie as actions change what is sensed in a complicated way. First, the agent only acquires $\mathbf{p}(t) = (\mathbf{x}(t), \mathbf{z}(t))$ recursively via the physical world, meaning that $\mathbf{p}(t+1)$ is not possible without getting $\mathbf{p}(t)$ first, $t = 1, 2, ..., n-1$ (e.g., an action must apply before one can sense the effect of the action). Second, each $\mathbf{x}(t)$ is a consequence of past physical experience $\{(\mathbf{x}(i), \mathbf{z}(i)) \mid i = 0, 1, ..., t-1\}$ (e.g., an object in $\mathbf{x}(t)$ is occluded partially by a moving arm while the arm moves up because of $\mathbf{z}(t-1)$). Third, each $\mathbf{z}(t)$ is a consequence of past physical experience $\{(\mathbf{x}(i), \mathbf{z}(i)) \mid i = 0, 1, ..., t-1\}$ (e.g., after receiving $\mathbf{x}(t-1)$ which indicates that the arm has reached the target object, the arm stops indicated by $\mathbf{z}(t)$).

6) Learning modes: Psychometrics is a field that studies experimental evaluation of the performance of a human using the fact that a human has a rich set of capabilities of autonomous learning. By *autonomous learning*, we mean that the brain inside the skull is fully autonomous from the time of conception. By *rich set*, we mean that the brain is able to handle any different modes of learning, from motor-supervised learning (called passive learning in psychology), to reinforcement learning (called instrumental conditioning in psychology), and to many other modes of communicative learning (called non-associate learning, cognitive learning, sequence learning, etc. in psychology).

7) Eight learning modes: Weng 2007 [93] argued that existing categorization of learning modes is not sufficient for modeling brain-like learning. [93] defined 8 types of learning, corresponding to all 8 combinations of three factors:(1) i: whether internal representation is handcrafted ($i = 1$ means yes and $i = 0$ means no), (2) e: whether the effector is supervised ($e = 1$ means yes and $e = 0$ means no) and (3) b: whether biased sensors are involved ($b = 1$ means yes and $b = 0$ means no). By biased sensor, we mean that the brain at birth time already has developed preference to its signals (e.g., pain receptor and sweet receptor). Therefore, the four learning modes with $i = 1$ belong to machine learning using symbolic models, while the other four modes with $i = 0$ belong to developmental learning (AMD) using emergent models. For the purpose of the theory here, it seems that the learning mode of emergent, effector-supervised, and communicative $(i, e, b) = (0, 1, 0)$ is the most basic and easy to understand. This learning mode is also used throughout this paper.

8) Experiments discussed in this work: This theory is based on embodiment and grounded mode of operation, regardless the effector is supervised (i.e., during training) or not (i.e., during practice). From the above discussion, we can see that we should not superficially consider that all embodiments must involve a real robot. For precise timing and evaluation of performance with the ground truth, we must use a simulated task environment but our experimental data include natural video and text from the Wall Street Journal. The TCM relies on real-time embodied sensorimotor experience. Without actions, the TCM learns nothing. Our video sequence is from a robotic setting and its action sequence simulates real-time robot-human interactions. Human is a part of the physical ground. Each text word is sensed as an image. Motor inputs-outputs are real-time, interactive and grounded. Real verbal actions (i.e., a talking vocal tract) from an emergent model are harder actions that roboticists need to investigate in the future.

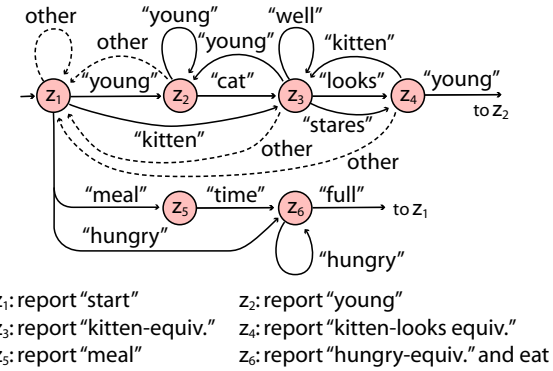


Fig. 2. An agent finite automaton (AFA) for a symbolic world. An FA can be extended to an agent which, at each time, inputs a symbol $\sigma \in \Sigma$ and outputs its state $q \in Q$. It starts from state z_1. From a state $q \in Q$, it takes an input $\sigma \in \Sigma$ and transits to a new state $q' \in Q$. A label 'other' means any symbol other than the symbols marked from the state. For example, state z_4 means the equivalent meaning of the attended last subsequence is "kitten looks" or equivalent. The "other" transitions from the lower part are omitted for clarity. The AFA does not include the text (e.g., "cat") for $\sigma \in \Sigma$ and the text for $q \in Q$ (e.g., report "hungry-equiv." and eat). Such meanings (text in natural language in this example) are only in the mind of human designer.

B. Handcrafted representations: SN

Some major differences among the traditional symbolic approaches to intelligence, many prior neural networks and the TCM are summarized in Table I. The columns of the table will be further discussed below, but we present the table early so that the reader can have a global picture when he reads on.

Finite Automata (FAs) [36] are one of the most popular symbolic temporal models that deal with time warping, sequential reasoning, and sequential actions. By "symbolic," we mean that the human designer hand picks the contents for, and handcrafts the boundaries of, the zones (modules) in the internal representations where each zone corresponds to an extra-body concept, as illustrated in Fig. 3. In an emergent model, however, there is no such static zones of extra-body concepts, since signals from sensors and effectors can potentially reach every neuron. That is, an emergent representation arises from receptors and effectors, not rooted in meanings of extra-body concepts.

By definition, a (deterministic) FA is a 5-tuple $(Q, \Sigma, q_0, \delta, A)$ where Q is a finite set of sates, Σ is a finite set of input symbols, $q_0 \in Q$ the initial state, $\delta : \Sigma \times Q \mapsto Q$ is the state transition function and $A \subset Q$ is the set of accepting states.

TABLE I
COMPARISON AMONG TEMPORAL METHODS

Methods	Neural network strengths		Symbolic network strengths		TCM new strengths	
	Representation	Incremental learning	Sequential abstraction	Immediate transfers	Un-modeled concept states	Un-modeled event states
Symbolic networks (SN)	Handcraft	No	Yes	Yes	No	No
Prior neural networks	Emergent	Yes	No	No	No	No
TCM (also brain)	Emergent	Yes	Yes	Yes	Yes	Yes

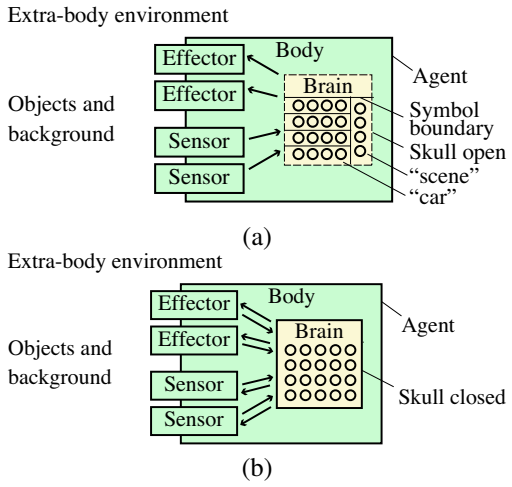


Fig. 3. Agents using (a) symbolic representation and (b) emergent representation. In (a) the skull is open during manual development (or this is an outside model for representations inside the skull) — the human programmers handpicked a set of task-specific concepts using human-understood symbols and handcraft the boundaries that separate concepts. In (b) the skull is closed during autonomous development — the human programmers only design task-nonspecific mechanisms for autonomous mental development and the internal representations autonomously emerge from experience. Many neural networks do not use the top-down connections in (b).

An FA with n states partitions all possible sequences Σ^* from an alphabet Σ into n classes, each represented by a state. Thus, each state defines an equivalent class of many input sequences, since all the sequences falling into a state q will be treated the same in the future, regardless how the sequence arrives at the state. Such a class is typically infinite in size because of various loops.

The classical definition of (deterministic) FA is for a language acceptor. The class of all FA corresponds to a particular category of languages, called regular language [36]. We need to extend the definition to agent FA:

Definition 1 (agent FA): An agent FA (AFA) M for a finite symbolic world is a 4-tuple $M = (Q, \Sigma, q_0, \delta)$, where Σ is the set of input symbols (alphabet), Q is the set of states for output, q_0 is the starting state, $\delta : Q \times \Sigma \mapsto Q$ is the state transition function.

An example of AFA is shown in Fig. 2. Since we consider states as also agent's outputs, each state contains a set of actions. A cognitive state can be considered a special case of action, as cognition can be reported by a reporting action.

It is important to note that the AFA does not "know" the meanings of input and output symbols, since the text that is associated with each input symbol $\sigma \in \Sigma$ and $q \in Q$ is only in the design documentation of the AFA that is understandable only by the human designers. In fact, the exact meanings of such documentation are not exactly the same across the minds of human designers. Humans are typically satisfied if such text documents appear to reach a consensus among human communicators. Namely, text symbols are consensual.

FA is the basis of many language processing systems, such as CYC, WordNet, and EDR [52]. Many cognitive models and knowledge-based models are also based on FA, such as ACT-R [2] and Soar [50].

FA has its many probabilistic variants, called SNs, to handle uncertainties using parametric learning, e.g., the Hidden Markov Model (HMM) [77], [78], Markov Decision Process (MDP) [75], Partially Observable Markov Decision Process (POMDP), [54], Markov Field (2-D version), and various Bayesian nets (also called belief nets and semantic nets) [83]. The handcraft nature is also true for these probability variants of FA. For example, a human designs a network composing of many HMMs in which each HMM detects a word (i.e., a symbolic meaning) [77] or an object [29] but the nodes within each HMM are subsymbolic. The meaning of a node in HMM is typically not rigidly determined, but through a pre-processing batch technique (e.g., k-mean clustering) before parameter refinement (e.g., using the Baum-Welch algorithm). However, the meaning of each HMM is predefined (e.g., represent a word "Tom"). For language processing, handcrafted syntactical rules have been used [12], [42].

What about other types of automata [36], such as Nondeterministic FA, minimum state FA, Pushdown Automaton, Linear-bounded Automaton, and Turing machine? All types of automata have been used to deal with primarily syntax in the traditional theories of language acceptors, not much semantics. For a language acceptor automaton, its language acceptance action is only in terms of syntax. Extended by the design documents as semantics associated with an AFA, the framework of AFA seems sufficient for modeling general purpose state-based symbolic relationships between input strings in Σ^* and symbolic actions in Q. In Section VII, we will see that the AFA theory is useful for modeling TCM external behaviors, but not sufficient for the internal representations of TCM.

C. Brittleness of input symbols

The high brittleness of an FA arises from the symbolic nature of input set Σ and output set Q. We first consider the symbolic nature of the input set Σ here. The symbolic nature of the output set Q will be considered in Sec. V-F.

The human designer needs to handcraft Σ to well represent all possible inputs to an acceptable precision. The number of inputs is intractably too large and handcrafting Σ is complex. If each input involves c concepts and each concept has v possible values, the potential number of input symbols is v^c, exponential in c. Suppose that we have $c = 22$ concepts and each concept has v=4 values (e.g., unknown, low, high, do-not-care), the number of possible input symbols is $v^c = 4^{22} = 16^{11}$, larger than the number of neurons in the brain. Here is an example of 23 extra-body concepts: name, type, horizontal location, vertical location, apparent scale, physical size, pitch, yaw, weight, material, electrical conductivity, shape, color, surface texture, surface reflectance, deformability, fragility, purpose, having life, edibility, usage, retail price, and owner.

Although the number of real-world inputs is infinite, a human designer only considers symbolic inputs such as the those above. Since the number is exponentially many, the human designer further manually groups these exponentially many original symbols into a tractable number of sets, each set being denoted by a symbol σ_i. This gives $\Sigma = \{\sigma_1, \sigma_2, ..., \sigma_n\}$, where each σ_i, $i = 1, 2, ..., n$, is a symbol, defined by a set of handcrafted conceptual conditions as feature set. However, the number of original inputs is exponential, too many to be checked exhaustively by the human designer for the validity of every input symbol σ_i.

This results in the well known high brittleness of symbolic inputs, regardless how the human designer (1) tries to iteratively improve handcrafted conditions for Σ or (2) uses probability (e.g., joint distribution of $(\sigma_1, \sigma_2, ..., \sigma_n)$). For (1), iterative improvements of the definition of Σ are ineffective since the human designer cannot check the validity of handcrafted conditions for each symbol σ_i because the number of original inputs v^c is exponential in the number of concepts c. For (2), probability will only reduce, but cannot sufficiently, the chance of errors in each symbolic input σ_i due to wrongly handcrafted conditions. The effectiveness of probability diminishes even when the number n of input symbols is moderate — the number of samples required to estimate joint distribution in $(\sigma_1, \sigma_2, ..., \sigma_n)$ is exponential in the number n.

For example, the weight concept of an apple can be disregarded in the Σ design if the current task is to find apples in a scene. However, such a Σ design breaks down when the robot needs to pack the found apples and move them because the weight concept becomes necessary. In other words, the high brittleness of SN is intrinsic, in terms of the design for its Σ.

Furthermore, such a static design for Σ is not viable for autonomous development, since the tasks that the agent will learn are unknown to the programmer before the agent "birth" and the internal representation of the agent is not accessible to the human programmer after the agent "birth" [103]. For example, the human programmer cannot simply design Σ as all the words in the Merriam-Webster Dictionary, since a human needs to learn new words in his life, beyond those in the Merriam-Webster Dictionary.

D. Emergent representations

The brain and artificial neural networks share the same characteristic in representation. Their internal representations emerge from learning experience, regulated by the genome or human programmed learning mechanisms [19], [76], [89], [66], [90].

Definition 2 (Symbolic and emergent): In a symbolic representation of an agent, each zone (often also called module) represents a *symbol* (e.g., text as label) about a concept about the *extra-body* environment. It is a human designer who handcrafts the *contents* of each zone and the rigid *boundaries* of the zones. An emergent representation does not allow such human handcrafted contents or boundaries, since the representations emerge through interactions between the brain and its external environments.

For example, many artificial evolutionary algorithms use a symbolic representation and they bypass development.

An emergent representation is harder for humans to understand, as it is *distributed* in the sense that each internal element (e.g., neuron) does not represent a pure linguistic meaning and a linguistic meaning is not necessarily constrained within a fixed subset of neurons. A neuron is typically involved in representing many meanings. Partially due to this distributedness, the internal representation of an area in the brain has been largely an open problem.

Many networks are feed-forward in operation (performance), e.g., Fei-Fei 2006 [22] and Serre et al. 2007 [84]. The Self-organization Maps [49], [64] were used mainly for unsupervised learning. Many artificial neural networks are also feedforward in operation, and use error back-propagation during a separate, non-operational learning phase [106], [51], [21]. Learning and operation are two different phases.

Other networks are recurrent in operation. Grossberg & Coworkers [30], [31], Deco & Rolls [15], Roelfsema & van Ooyen [81] have used top-down connections in their networks. The Wake-Sleep algorithm [33] and the Deep Belief Nets [34] used unsupervised learning. The bottom-up and top-down connection weights are "tied" and they are learned using a greedy algorithm through up-pass phases and down-pass phase. Sit & Miikkulainen [85] and Weng et al. [101], [102] used top-down connections in their laterally interactive self-organization networks. Inspired by the top-down connections in the laminar cortex, the Multi-layer In-place Learning Networks (MILN) [101], [102] was originally proposed as a model for cortical spatial pattern recognition, instead of processing temporal information. MILN is the early spatial cortical model for TCM.

MILN and TCM do not use error back-propagation, to avoid the lack of long-term memory in error back-propagation methods. Like the cortex, the motor signals are directly projected into early areas for internal self-organization, because action errors are not available during animal's autonomous development (e.g., trials and practice).

Interestingly, as the motor signals can be imposed by teachers at will, this learning model allows semi-supervised learning. As soon as the motor port is not supervised by the teacher, the network generates its own motor signals for practice for autonomous learning without supervision.

Furthermore, the learning method in MILN and TCM is *in-place*. The *in-place* learning concept is more strict than the well known but loose concept of local learning. By *in-place*, we mean that each neuron is responsible for its own learning and computation, and there is no need for any extra-cellular mechanisms that many local network learning methods require. For example, there is no need for computing the cross-correlation matrix for the input vector of every neuron. This is important for both the biological plausibility and the brain-size tractability in engineered systems.

E. Emergent representation with states

A state represents a temporal context in a system, like an FA. It should represent spatiotemporal context in general, but this work focuses on temporal processing.

As reviewed by Ivry & Schlerf [40], there are two frameworks with regard to how the brain (or a network) deals with time, one using dedicated temporal components and the other using a network without dedicated temporal components.

In the first framework, called time-dedicated framework, the main scheme for longer time dependency is to create a sequence of delay units. Examples of such scheme include Hochreiter & Schmidhuber 1997 [35], Lin et al. 1996 [53], Wiemer 2003 [107], and Drew & Abbott 2006 [17]. All these models explicitly model time in the internal representation and network behaviors.

In the second framework, called time-nondedicated framework, the temporal behaviors of brain (or network) are considered an emergent outcome of intrinsic cellular mechanisms. We discuss the second framework below.

F. Emergent time-nondedicated framework

Intracellularly and extracellularly, the concentration of a type of molecules (e.g., morphogens, neural transmitters, and neural modulators) takes time to build, and the concentration tends to dissipate spatially through time. Such graded temporal mechanisms are involved in the operation of a cell, as well as the development of tissues, organs, and the wired structure of the brain. They arise from the interactions of multiple cells, each using its intrinsic intracellular mechanisms. Nevertheless, they seem unlikely the major players in generating fast-changing brain responses that have a long-term dependency on environmental events.

Two types of models have been proposed within the time-nondedicated framework: (1) the internal state type where states are not directly observed and shaped by the external environment; (2) the external state type where states are directly observed and shaped by the external environment.

The internal state type includes randomly connected units in a hidden area, such as the Buonomano-Merzenich network [9], LSM [61], LSTM [35], ESN [41], and Reservoir Computing [60]. The basic idea is that a randomly and recurrently connected internal hidden area H should respond sequentially to an input sequence. If every different moment of the state sequence is represented by a unique firing pattern of H, the firing pattern of H should can represent all temporal information of the input sequence without ambiguity. The output area can then "read out" the firing pattern of H using a logic-OR like mechanism. However, as we will discuss in Sec. V-F, the number of sequences is exponential in the temporal length of the input sequence. A finite number of neurons in H causes some different sequences to have a very similar firing pattern, causing unpredictable "collapses" (or near "collapses" in the numerical space) between sequences that require different action outputs. The longer the required time dependence, the more "collapses" may occur.

Another problem is also severe: Such internal states are "concrete" — depending on actual sensory forms instead of actions. Such internal states do not "collapses" when they should, different from the recurrent abstract states of FA which enable skill transfers discussed in Sec. VII-F.

G. The trap of internal interference by human designer

For the above reason, LSTM [35] and oscillator-based models [67] introduce task-specific (handcrafted) logic-like mechanisms into internal hidden area H to maintain some handcrafted short-term memories over long time periods; but this belongs to manual internal interference by the human designer, resulting in a task-specific symbolic network, although some components of the area H are emergent.

H. Emergent and time-nondedicated: External states

DN is an emergent and time-nondedicated brain model [97], [98]. TCM is its temporal name. The states of TCM are however externalized — open to the external environment. Since the external states are observable, teachable and shapeable by the external environment, TCM allows external supervision (which is emergent too!) to recursively and incrementally map different sequences into the same state, analyzed using the FA theory here. Interestingly, such external supervisions arise not only from the external human teachers and the deeper causality of the physical external world (e.g., when the earth shakes, it shakes me too), but also from the brain's own actions resulting in autonomous self-learning (e.g., trials and errors). Thus, TCM is a temporal model that both the external environment and the internal environment (i.e., the brain itself) can teach.

Yet, no neuron or any module inside TCM has a dedicated identification with respect to time. For example, TCM can maintain or disregard temporal information dynamically depending on context. For example, in a context the time duration information must be disregarded (e.g., text reading discussed below); but in another context the time duration information must be precisely counted (e.g., perceiving the duration of a visual event [40]).

As far as the authors know, the TCM here is the first published emergent method that has demonstrated not only long term dependency in behaviors but also incorporation of both duration-insensitive behaviors and duration-sensitive behaviors.

IV. Temporal Mechanisms

This section provides a conceptual overview of existing mechanisms for temporal processing before presenting the

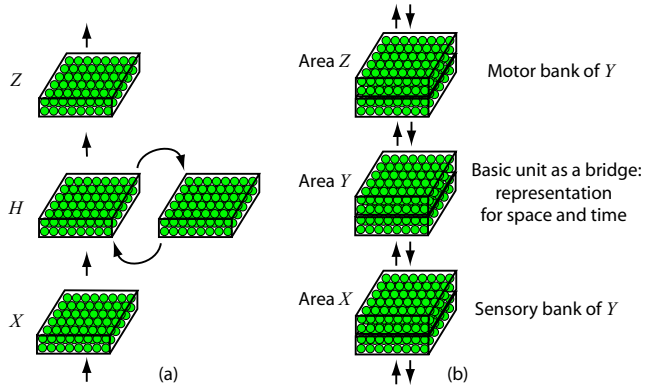


Fig. 4. Two types of temporal architectures using emergent representations. (a) Hidden feedbacks: the hidden area H has local feedbacks. (b) Bridge with sensory and motor banks: Each cortical area Y serve as a "bridge" that helps to predict the signals in its two banks X and Z.

concept of the TCM new temporal mechanism. They are mechanisms used by some practical models.

All temporal mechanisms fall into two categories, symbolic and emergent. In the first category, the unit of temporal windows is symbolic in nature. In the second category, there are two existing temporal approaches, (i) temporal windows and (ii) feedbacks that are local and hidden. The new temporal mechanism of TCM does better than all those existing mechanisms. It belongs to the emergent category, and it is a new temporal approach: (iii) feedbacks that are global and open. Shown in Section VII, the new TCM mechanism is logically as clean as symbolic models.

We first discuss, in the following three subsections, major existing techniques to deal with time.

A. Background: Symbolic state-based models

An FA with word labels as inputs is shown in Fig. 2. It drops stop word "well," to map phrases of different lengths to equivalent context states. For example, state z_4 indicates that a legal noun phrase (NP) followed by a verb phrase (VP) of an equivalent class (meaning "kitten looks" or the alike) has been detected. However, this FA is static once handcrafted. It cannot deal with new words or new sentences [103]. Thus, this temporal mechanism is not suited for autonomous development.

B. Background: Temporal windows in emergent models

An intuitive way to deal with time is to use a temporal window of a certain length n as a working memory [93]. However, such an intuitive way cannot deal with time of arbitrary length and time warping. The number of cases observable in a temporal window of length n is exponential in n: $\mathcal{O}(k^n)$, where k is the size of the vocabulary. Each eye has only one retina which cannot store n images for a moderate n. This mechanism is not scalable to long temporal context.

C. Background: Hidden feedbacks in emergent models

For a neural network, we denote X as its vector input port and Z its vector output port. A major temporal mechanism for output Z to depend on multiple frames in X is to create a local feedback as illustrated in Fig. 4(a).

Examples of such local recurrence include the Hopfield network 1982 [37] (X-to-X feedback where $X = Z$), the Boltzmann machine 1985 [1] (the stochastic counterpart of Hopfield network), the Jordan Network 1986 [46] (Z_d as Z node-wise delay, Z_d node-wise recursive delay loops, and H takes input from X and Z_d), and the Elman Network 1990 [18] (Y_d as Y node-wise delay, and H takes input from X and H_d). Such local loops provide a context of only a limited temporal length, since the temporal context fades away exponentially as the delayed vector is recursively mixed with the current inputs.

For a longer time dependence, the Buonomano-Merzenich network 1995 [9], LSTM [35], LSM [61], ESN [41], CTRNN [109], and the Reservoir Computing [60] extended the hidden area H to a large randomly locally connected recurrent area. The memoryless output port Z reads the response from the hidden network H in a task-specific fashion through an explicit search [9] or an implicit search via a gradient-based technique [61], [41] for H-to-Z connection weights. RTRNN [109], [72] further used two different subareas of H, one with a fast, and another with a slow, leaky current of the unit's membrane potential. Since the hidden H network is randomly generated and has a larger number of neurons, the H network serves as a separator of temporal sequences from X. Such a separator suffers from the memory fading problem — the current firing pattern in H depends increasingly less on older values in X. LSTM [35] and oscillator-based models [67] handcrafted task-specific mechanisms in the network H so that certain short-term temporal memory can be kept for as long as the task needs. Reservoir Computing [60] extended the ideas of the above networks by considering the hidden network H as a reservoir, which is randomly connected and can also be learned using some gradient-based techniques.

We will see in Sec. V-B that the number of sensory sequences of length n that must be distinguished is exponential in n. Since the above schemes require the hidden area H to learn in an unsupervised way, an exponential number of sensory sequences needs to be distinguished by H, which is impractical for a task of a moderate sequence length.

Another major limitation of such hidden-area schemes is the lack of generalization power, because of the absence of systematic state equivalence and state-based skill transfer.

D. New emergent model: Open and global motor feedbacks

Our basic TCM architecture theory is based on the following scheme. The concepts of sensor and motor are relative. If two areas A and B are directly connected to each other bi-directionally and A is closer to sensors and B is farther, B is then the motor area of A and A is the sensory area of B. Fig. 4(b) shows a basic unit of TCM, with its internal area Y bidirectionally connecting with its sensory area X and its motor area Z.

The above TCM architecture is based on extensive neuroanatomical studies available as early as early 1990s [24], [108], [10]. Almost all connections between two brain areas

are bi-directional. Both types of feedbacks exist, cortical feedbacks (within a cortical area) and cortico-cortical (between cortical areas).

Cortical feedbacks (within a cortical area) have been modeled by the LCA [100] theory, which models how a cortical area A deals with its cortical connections within its area while taking its cortico-cortical inputs from other areas. As shown in Fig. 4(b), the input space to Y area is the space $X \times Z$, defined as $\{(\mathbf{x}, \mathbf{z}) \mid \mathbf{x} \in X, \mathbf{z} \in Z\}$.

Cortico-cortical (between cortical areas) feedbacks have been reported by extensive neuroanatomical studies [24], [108], [10]: Later cortical areas extensively feedback to earlier cortical areas. For example, not only V1 connects to V2 in both ways, but also V2 connects to V3 in both ways. Further, V1 connects to V3 also in both ways, although the amount of connections is relatively minor, because, we predict, that the statistical correlation between V1 and V3 are not as strong as those between consecutive areas.

We use the intuitive term "brain" to refer to the central nervous system, which includes the spinal cord, the lower brain, the mid brain and the forebrain. These different brains deal with different sensory and motor modalities, as well as their integration. For example, the spinal cord at Y area handles the somatic sensory port as its sensory bank and the limb effector port as its effector bank. The forebrain seems to integrates all sensory modalities (e.g., visual) and all motor modalities, while taking all sensory ports, all lower brains, and all effector ports as its banks.

Therefore, in general, we model a brain area Y (e.g., a Brodmann area [47, pp. 325-328]) to have its sensory area X and its motor area Z as illustrated in Fig. 4(b), regardless where it is in the brain, the spinal cord, cerebellum, brain stem, or cerebral cortex. Often, it is not obvious which area is closer to sensors, e.g., an area (e.g., LIP) between two pathways (e.g., the dorsal and ventral pathways). As we will see, the learning mechanisms for two connected banks X and Z are largely symmetrical. Thus, between X and Z, which is sensory and which is motor is immaterial.

If the sensory area X is the retina and the motor area Z consists of all the muscle neurons, the basic cortical unit Y is the entire brain. The detailed structure in Y emerges to reflect the statistics of the signals in X and Z. The more resources the brain Y has, the better approximation the two-way mapping can predict between its sensory area X and its motor area Z, and thus, more sophisticated capabilities the brain can acquire.

For simplicity in understanding and in experiments, we will use a *canonical* symbol-vector mapping: Each symbol is mapped to a different neuron in X and Z. Thus, in the mind of human observer, the highest responding neuron in X and Z represents the corresponding symbol. However, this canonical representation is wasteful. In general, multiple motor neurons are allowed to fire concurrently in X and Z. For a d dimensional space X, the number of different binary patterns is 2^d, exponential in d. Therefore, the input area X can represent virtually any practical number of inputs. For Z, different muscles can be active concurrently to generate complex action sequences.

As far as we know, the scheme of motor-assisted temporal abstraction further analyzed below seems the first computational model that explicitly models how the spatial brain deals with time *without* explicit mechanisms that are dedicated exclusively to time. The early idea and preliminary experimental results where first published in 2008 [58] and 2009 [104], [105].

V. Analysis of the Temporal Problems

This section provides deeper insight into the challenges of temporal information processing and the daunting computational complexity that a general-purpose temporal brain faces. It explains why prior temporal mechanisms are not as effective as TCM from an analytical point view.

We need to first discuss the modalities of sensory input first so that the analysis is applicable to various sensory modalities.

A. Generality for sensory modality

A sensory input or a motor output at any time t can be considered an image, a receptor firing pattern in retina, cochlea, or skin. A motor response vector is also an image of the responses from an array of muscle neurons. An image is represented as a vector, where components are indexed either along a 1-D axis, on a 2-D plane, or in a 3-D volume. The brain-inspired TCM is applicable to any sensing modality and any effecting modality, as long as the sensory port and the motor port are properly defined for the agent.

The same is true for a text input modality, e.g., printed or hand-written. However, unlike traditional processing techniques that treat text inputs as a sequence of symbolic words, TCM treats such inputs as a sequence of temporal images, like an eye scanning text in time. For example, each word may be "fixated on" by the "eye" of TCM for a duration that spans multiple discrete time frames. This consideration is important to deal with the well known problem of temporal time warping: At different times you read a given page, your eye scans the text on the page with a different speed.

For simplicity, we assume electronic word inputs, through which a unique electronic code is received representing a predefined word. An English sentence s is composed of a series of word labels: $s = (w_1, w_2, \ldots, w_l)$, where l is the length of the sentence. A space after a word is a word terminator, but more than one consecutive space are treated as a single space. All normal English punctuation marks are also treated like words. Each word is represented as an image.

B. Recursive temporal abstraction for sensory inputs

What representations should TCM generate internally when it receives one word at a time? At one extreme, every *subsequence* from the first word of a book corresponds to a different context state and needs a different motor action. This type of representation is not very useful as such a state is hardly shared by another experience. At another extreme, every *single word* corresponds to a different internal state and a different motor action regardless the words preceding it. This type of representation is not powerful either as the machine is only a word-based reflex agent. It cannot make sense from a

sequence of multiple words. Thus, we must consider an input sequence of an arbitrary length but enable a temporal action to be generalizable.

Suppose that a temporal action is based on n image frames, $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ where $\mathbf{x}_i \in X$, $i = 1, 2, ..., n$, and $X = R^d$.

A direct batch estimation of probability density of this sequence deals with joint probability density $p(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)$ in (nd)-dimensional space directly. This is intractable for even a moderate n because of the exponential complexity: Suppose that there are m different vectors in X, the number of sensory sequences of length n in the form of $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ above is m^n, exponential in n. In Sec. III-C, we have derived that the number of potential input symbolic objects in X is $m = v^c$. Then, the number of different sensory sequences of length n is $m^n = (v^c)^n = v^{cn}$, growing exponentially in cn.

In a recursive manner, we can factor the probability density

$$p(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n) = p(\mathbf{x}_1) \prod_{i=2}^{n} p(\mathbf{x}_i \mid \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{i-1}).$$

This is recursive estimation of temporal distribution of sensory inputs. However, it is still impractical to estimate the conditional probabilities $p(\mathbf{x}_i \mid \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{i-1})$ because the number of sequences in the form of $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{i-1}$ is $v^{c(i-1)}$.

In syntactic language processing [42] and in temporal Bayesian networks [91] it is typical to hand label the equivalent class of word string $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{i-1}$ as equivalent temporal state $\phi(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{i-1})$ so that the probability above is equivalently replaced by

$$p(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n) \approx p(\mathbf{x}_1) \prod_{i=2}^{n} p(\mathbf{x}_i \mid \phi(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{i-1})).$$

This corresponds to recursive estimation of temporal distribution using recursive abstraction of sensory inputs — from many concrete sequences in $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{i-1}$ to a single abstract class $\phi(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{i-1})$.

Many existing formulations of (symbolic and connectionist) language processing models belong to this framework (e.g., [20] and the review therein), in the sense that they are classifiers for input sequences.

C. Abstraction from input batch to the action

The behavior is the major goal for the brain, instead of its representation of sensory space. Many parts of sensory inputs (e.g., the exact duration of each syllable or stop words) are not relevant to actions. With the TCM model, the major goal of development is to produce the most likely actions $\mathbf{z}_n$, a vector for actions, that are appropriate for the agent age group. That is, it is the *behavioral* distribution

$$p(\mathbf{z}_n \mid \phi(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{n-1})), n = 1, 2, ... \quad (7)$$

that is the focus of development, instead of the estimation of the higher dimensional distribution of *sensory inputs* $p(\mathbf{x}_i \mid \phi(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{n-1}))$.

The following aspects further motivate the above formulation.

First, motor actions can be externally observable, but the same is not necessarily so for the temporal abstraction $\phi(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{n-1})$ for sensory inputs, as the agent is "skull-closed". Not all the details in $\phi(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{n-1})$ are necessary for the desired motor actions from an agent at certain age. Only information (e.g., excluding the absolute vector length) that are necessary for generating actions should be considered.

Second, the action $\mathbf{z}_{n-1}$ can be abstract. Each action vector $\mathbf{z}_{n-1}$ from TCM is one of many instances of the abstract class (e.g., reporting a class label). The motor action $\mathbf{z}_{n-1}$ lumps many different but equivalent spatiotemporal input sequences of various temporal lengths (e.g., various ways of expressing "jealousy") into a single instance $\mathbf{z}_{n-1}$ that belongs to the abstract class (e.g., saying "jealousy"). This is because TCM produces different action values almost all the time, even with the canonical motor representation. For canonical motor representation, e.g., we consider the abstract action as the component that has the highest response value in the action vector $\mathbf{z}_{n-1}$. Thus, although each action instance itself is concrete, in the mind of the human observer each action is abstract, representing the abstract meaning of the highest responding motor neuron (muscle).

Third, each action $\mathbf{z}_{n-1}$ can represent any human communicable abstract concepts. All such concepts are coded abstractly by a human language (e.g., "anger" or "jealousy") and can be said, written and signed though motor actions.

Fourth, part of each action $\mathbf{z}_{n-1}$ can be covert. When you do not want other to hear your voice, you can say very softly so that only yourself hear it as a rehearsal. However, your overt action is still going on. Thus, TCM can privately rehearse using its covert part of actions.

D. Recursive action abstraction

Eq. (7) corresponds to batch processing, for every sample time, $t = 1, 2, ...$ The brain cannot collect and keep all inputs from the birth $\mathbf{x}_1, ..., \mathbf{x}_{n-1}$ and produce an action $\mathbf{z}_n$.

However, as discussed above, $\mathbf{z}_{n-1}$ can be abstract and sufficiently rich so that $\mathbf{z}_n$ can use $\mathbf{z}_{n-1}$ as attended temporal context instead of the intractable $\mathbf{x}_1, ..., \mathbf{x}_{n-1}$. The TCM network recursively converts an intractable problem on the left below to a tractable one on the right:

$$\max_{\mathbf{z}_n \in Z} p(\mathbf{z}_n \mid \mathbf{x}_1, ..., \mathbf{x}_{n-1}) = \max_{\mathbf{z}_n \in Z} p(\mathbf{z}_n \mid \mathbf{z}_{n-1}, \mathbf{x}_{n-1}) = \mathbf{z}_n^*$$

where $\mathbf{z}_{n-1}$ recursively abstracts like $\mathbf{z}_n$. Although such an agent is *reflexive with states* [83], the abstract nature of $\mathbf{z}_{n-1}$ makes this much simpler agent to be equivalent to the very complex agent on the left side. Of course, the learning of $\mathbf{z}_{n-1}$ is critical for the effectiveness.

E. Prediction for both actions and inputs

Recursively, the TCM treats the best $\mathbf{z}_n^*$ as the action to be attended in this context.

$$\mathbf{z}_n^* = f_z(\mathbf{z}_{n-1}, \mathbf{x}_{n-1}) = \max_{\mathbf{z}_n \in Z} p(\mathbf{z}_n \mid \mathbf{z}_{n-1}, \mathbf{x}_{n-1}) \quad (8)$$

where $f_z : Z \times X \mapsto Z$ is the action prediction function for the continuous spaces X and Z. It is important to note that the action $\mathbf{z}_n^*$ includes also the cognitive state, as illustrated in Fig. 2.

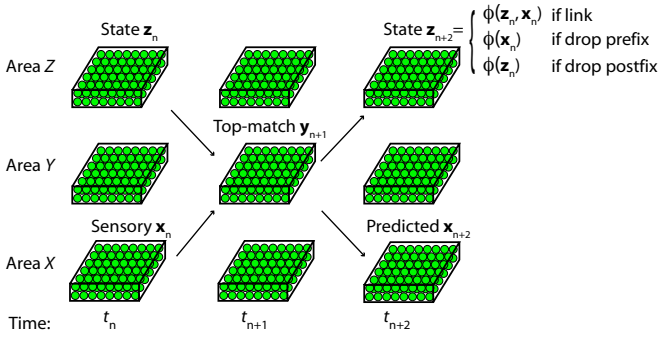


Fig. 5. Sequential decision making or spatiotemporal motor action using framewise motor-assisted temporal abstraction by a basic 3-area unit TCM. At each temporal unit shown above, three basic operations are possible: link, drop prefix, and drop postfix. After proper training, the TCM is able to attend any possible temporal context.

We call this temporal scheme, *the framewise motor-assisted abstraction* scheme for the prediction of $\mathbf{z}_n$. It uses a spatial network to deal with general temporal contexts.

Likewise, the prediction can also be done for $\mathbf{x}_n$ to predict the attended part in the sensory input $\mathbf{x}_n$, to relatively suppress the unattended part in $\mathbf{x}_n$, and to reduce the noise in the attended part in $\mathbf{x}_n$:

$$\mathbf{x}_n^* = f_x(\mathbf{z}_{n-1}, \mathbf{x}_{n-1}) = \max_{\mathbf{x}_n \in X} p(\mathbf{x}_n \mid \mathbf{z}_{n-1}, \mathbf{x}_{n-1}), \qquad (9)$$

where $f_x : Z \times X \mapsto X$ is the sensory prediction function for the continuous spaces X and Z.

Now, considering continuous time but sampled at discrete times, $t = 0, 1, 2, ...$, mathematically, Eqs. (8) and (9) combined give the temporal function below.

$$X(t-1) \times Z(t-1) \xrightarrow{f} X(t) \times Z(t). \qquad (10)$$

where f is a recursive function for temporal processing where Z has a full freedom of representation — representing the necessary, often abstract, spatiotemporal state which can be supervised by the external and internal environments.

Furthermore, as we will see, the response function f of TCM approximates the underlying probability, because the optimality of $\mathbf{z}_n^*$ and $\mathbf{x}_n^*$ as the next $\mathbf{z}_n$ and $\mathbf{x}_n$, respectively. The TCM model relates to probability variants of FA, such as HMM, MPD, POMDP, and Bayesian nets.

However, the internal representations in a TCM are emergent instead of handcrafted. As we will see next, because internally TCM is based on top winners from competition, it does not need to require that all the responses sum to 1, an idea also used in fuzzy logic. This more flexible framework is inspired by the lateral inhibition mechanisms found in the cortex. Olshausen & Field [69], [70] proposed that the cortex uses sparse coding in the sense that few neurons in each cortical area fire at any time. Our argument here, based on the LCA theory, is that sparse coding is not only useful for generating local features (i.e., only a part of the receptive field is non-zero) as argued by Olshausen & Field, but more critically, necessary for allowing all other neurons do not fire and update so that their long-term memories are kept.

At this point, our analysis has not addressed the internal representation yet. Let us analyze what will happen if we do not use internal representation.

F. Brittleness of symbolic states

Using symbolic networks (SNs), a human handcrafts samples in X and Z as symbols in Σ and Q, respectively. Using such symbolic coding, Eq. (8) corresponding to the state transition function of a handcrafted SN. An SN does not predict input symbols in Σ and therefore it does not do f_x in Eq. (9).

Suppose that we need to handcraft an SN to understand text in a natural language [4]. Each $\mathbf{x}_i$ in $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{n-1})$ is classified by a symbolic word σ. For simplicity, we consider FA first. Then, Eq. (8) becomes deterministic. Suppose that each $\mathbf{x}_i$ represents an typewriter symbol, one of "A" to "Z" and punctuation symbols, etc. However, this means that the FA has to deal with misspelled words, desirable but intractable by our human designer. Instead the human designer handcrafts only meaningful strings that form valid English words. Eq. (8) becomes

$$q(t_n) = \delta(q(t_{n-1}), \sigma(t_{n-1}))$$

where we use symbolic input σ to classify the real vector input $\mathbf{x}$, the symbolic state q to classify the real vector output $\mathbf{z}$, and δ to denote the symbolic function corresponding to f_z. We will see in the next subsection that the above expression corresponds to the transition function $q' = \delta(q, \sigma)$ of FA.

Then, the handcrafted state q must contain all the necessary information of the equivalence label $\phi(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{n-2})$ so that q and σ are sufficient for δ to find the unique q'.

We now consider the brittleness of Q of all SNs discussed in Sections III-B. A human manually merges multiple symbolic sequences in the form $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{n-2}$ into each handcrafted symbolic state $q = \phi(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{n-1})$ in Q, so that Q becomes a set of output states.

As we discussed in Sec. III-C, the number of different symbolic inputs in $\mathbf{x} \in X$ is $m = v^c$ for c concepts. The number of sensory sequences in the form of $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{n-1}$ is then $m^{n-1} = (v^c)^{(n-1)} = v^{c(n-1)}$, exponential in $c(n-1)$. Therefore, it is intractable for a human designer to check the consistency of every $q \in Q$, for all the $v^{c(n-1)}$ sensory sequences. This corresponds to the high brittleness of an SN in terms of its handcrafted symbolic states in Q.

HMM and POMDP use probabilities for $q \in Q$ to alleviate the problem with wrongly handcrafted states. Nevertheless, as we discussed earlier for input symbols, probability in terms of q can only reduce but cannot eliminate errors in each inappropriately handcrafted state q.

Therefore, due to the brittleness of Σ and Q, the high brittleness of SNs is intrinsic, seemingly unsolvable.

G. Collapse of sequences in unsupervised hidden area

The exponential number of sequences in $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{n-2}$ above also account for the lack of sequence generalizability of all prior temporal neural networks reviewed in Sec. IV-C, such as Hoffman nets, Jordan nets, Elman nets, LSM, LSTM,

ESN, and Reservoir Computing. Regardless whether the internal hidden area H is randomly constructed, handcrafted, or learned using a gradient based method, the number of temporal sequences in the form $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{n-2})$ that must be distinguished is exponential in the length $n-2$. Suppose $\mathbf{x}$ is represented by symbolic σ, and each σ is a word. Suppose that Σ contains 1000 English words, the number of all possible 4-word phrases in the form of $(\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ is $1000^4 = 10^{12}$, more than the number of neurons in the brain. The human designer must partition all 10^{12} 4-word phrases into a moderate number m of states. Again, the human designer is not able to check all these 10^{12} 4-word phrases to be meaningful ones and meaningless ones (to output to a state q representing "nonsense sentence"), leading to the brittleness of states in Q.

H. Using internal representations

Eq. (10) does not address whether internal representation is used.

An FA does not use any internal representation, as its Σ and Q are all handcrafted, static and rigid. Because of the inconsistencies observed in real applications, HMM and POMDP extended from FA so that their state transition $\delta : Q \times \Sigma \mapsto Q$ becomes probabilistic. Bayesian nets are special forms in the sense that nodes have additional symbolic conditions of other nodes. The probability values of state transitions are learned based on the handcrafted boundaries inside the FA base. In such SNs, there are two layers of probabilities, those of each $\sigma \in \Sigma$ and those of each $q \in Q$. The HMMs, POMDP, and Bayesian nets typically treat the uncertainty of $\sigma \in \Sigma$ as that of $q \in Q$ (hence the term "hidden").

Fig. 5 illustrates how TCM uses internal representation as Y area to conduct this two-way prediction for both X and Z.

The concept of internal representation of TCM does not apply to SN because SNs are not developmental. Although the internal connections of TCM have close relationships with probability as we will see, they are for components of emergent vector representations in general, instead of a symbolic concept of the external environment.

Only developmental agents have "skull" closed network (e.g., brain or network) whose development is autonomous throughout the lifetime. For example, the cortical areas X, Z and Y of TCM use distributed representation which is dynamic and adaptive. The areas X and Z have their external ports open to the environment, The Y area and its connections with X and Z are internal representations inside the "skull".

Computationally the expression in Eq.(8) is achieved by in-place computation of three areas X, Y and Z. Suppose that Y bi-directionally connects X and Z. We consider Y a bridge between X and Z. From each input from both sources $\mathbf{p} = (\mathbf{x}, \mathbf{z}) \in X \times Z$, the area Y uses is emergent memory $V = (\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_c)$ to compute its response vector $\mathbf{y}$ as a representation of the top-k matches among V:

$$\mathbf{y} = f[\text{top-k} \max_{1 \le i \le c} r(\mathbf{v}_i, \mathbf{p})] \tag{11}$$

where $r(\mathbf{v}_i, \mathbf{p})$ is the pre-response which measures the goodness of match between $\mathbf{v}_i$ and $\mathbf{p}$, and f is a dynamic nonlinear function [100] that takes top-k pre-responses and maps each to a standard response range $(0, 1]$ according to its rank among the top-k winners so that $\mathbf{y}$ is a sparse vector — only a small number k of its components in $\mathbf{y}$ are non-zeros and the top one neuron always give a response value 1. Such a basic cortical unit is illustrated in Fig. 1. The top-k like competition is probably realized by lateral inhibition among neurons through fast network updates. Using slower software, we take advantage of top-k sorting to find the top-k winners in each network update. This shows an advantage of digital computer, although top-k sorting itself is not in-place.

Let us consider text sequence. Given a sentence $s = (w_1, w_2, \ldots, w_l)$, the TCM scans one frame (or word) w_i at a time, learns and operates incrementally. Occasionally, its actions are supervised. During the testing phase, the adaptive part of the network (weights and neuronal ages etc.) can be fixed to avoid update during testing.

Consider the three-area network in Fig. 1, running at discrete times $t = 0, 1, 2, ...$ This is without loss of generality, as well known in digital signal processing — an appropriate sampling rate can always sample any continuous flow to a required finite precision. The area Y takes the top-down input $\mathbf{z}(t-1)$ from Z as the top-down temporal context and the bottom-up input $\mathbf{x}(t)$ from X which represents the current image or word. Its area function implemented by LCA maps $\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t)$, based on its area memory $N_y(t)$, to its response $\mathbf{y}(t+1)$ and updates the area memory to $N_y(t+1)$:

$$(\mathbf{y}(t+1), N_y(t+1)) = f(\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t), N_y(t))$$

where f indicates the generic area function in Eq. (1). The areas X and Z compute in the same way.

During the next network update, area Z takes bottom-up input $\mathbf{y}(t+1)$ from Y, based on its area memory $N_z(t+1)$, to its response $\mathbf{z}(t+2)$ and updates the area memory to $N_z(t+2)$:

$$(\mathbf{z}(t+2), N_z(t+2)) = f(\mathbf{y}(t+1), \mathbf{z}(t+1), N_z(t+1))$$

where f indicates the generic area function in Eq. (1). If the teacher wants to supervise the motor, impose the desired value $\mathbf{z}(t+2)$. The areas X and Z compute in the same way.

Viewed internally, each area Y update realizes $X \times Z \mapsto Y$ internally, there X and Z are its two connected areas. Viewed externally, two network updates realize not only the forward mapping $X \times Z \mapsto Z$ for action generation but also the backward mapping $X \times Z \mapsto X$ for attention selection.

This 3-area TCM seems to be applicable to a generic brain area. For example, if X is V1 and Z is V3, then Y is V2. If X is all receptors and Z is all the motor neurons and glands, Y is the entire brain (central nervous system). However, this is a theoretical computational prediction. Extensive neuroscience studies are needed to verify this prediction.

I. Temporal attention

At each frame time, there are three basic operations of temporal attention as illustrated in Fig. 5, determined by what is learned at motor output $\mathbf{z}(t+2)$:

1) **Link**: If $\mathbf{z}(t+2)$ represents the context $\mathbf{z}(t)$ followed by $\mathbf{x}(t)$, the network "links" contexts to make the temporal

context longer. For example, if $\mathbf{z}(t) = \phi(abc)$ and $\mathbf{x}(t) = d$, then $\mathbf{z}(t+2) = \phi(abcd)$, linking the class of abc with the class of d.

2) **Drop prefix:** If $\mathbf{z}(t+2)$ represents the equivalent class of $\mathbf{x}(t)$, the network "drops" the prefix. For example, if $\mathbf{z}(t) = \phi(abc)$ and $\mathbf{x}(t) = d$, then $\mathbf{z}(t+2) = \phi(d)$, dropping the prefix abc.
3) **Drop postfix:** If $\mathbf{z}(t+2) = \mathbf{z}(t)$, the network "drops" input $\mathbf{x}(t)$ as it keeps the last context $\mathbf{z}(t)$. For example, if $\mathbf{z}(t) = \phi(abc)$ and $\mathbf{x}(t) = d$, then $\mathbf{z}(t+2) = \phi(abc)$, dropping the postfix d.

Fig. 5 indicates that it takes two network updates for the effect of $\mathbf{x}(t)$ to start to show up at the motor end as $\mathbf{z}(t+2)$. When the teacher likes to supervise the action at the motor end, he should consider this effect.

J. Observations

Unlike the " bag of words" approaches where the order of words is not considered to avoid otherwise the exponential complexity, we will use the text scan mode and reduce the exponential complexity to linear complexity. As far as we know, TCM is the fist system that uses actions to abstract for fully automatically generated spatiotemporal internal representations. In natural systems, such actions are not always correct, but can be corrected through interactions. This is in contrast with, e.g., HMM based speech recognition methods which use a static handcrafted structure of multiple HMMs.

VI. Functions of Representation

Unlike an SN, TCM uses emergent representation as shown in Fig. 4(b). Instead of a single symbolic state at any time as illustrated in Fig. 2, a 3-area TCM has 3 levels of distributed representation, one for each area $A \in \{X, Y, Z\}$.

A. Internal neurons as soft AND of X and Z

Given any input pair $\mathbf{p} = (\mathbf{x}, \mathbf{z})$, LCA finds the top neuron(s) who gives the highest pre-response $r(\mathbf{p}, \mathbf{v}_j)$ (i.e., best matching). Thus, the best matched neuron j serves as the representative of unknown input $\mathbf{p}$. As indicated in Eqs.(2) and (3), both components of $\mathbf{x}$ and $\mathbf{z}$ must match well with top-k $\mathbf{v}$'s:

$$\mathbf{p} \approx \mathbf{v}_{j_1}, \mathbf{p} \approx \mathbf{v}_{j_2}, ..., \mathbf{p} \approx \mathbf{v}_{j_k}.$$

Thus, the Y area serves as a soft AND: All the corresponding components in $\mathbf{x}$ and $\mathbf{z}$ must match well with the top $\mathbf{v}$. This soft AND is due to (1) there is a sufficiently large number c of neurons in Y and (2) that the response of Y is sparse (i.e., k/c is very small), so that only the best matched neurons can fire.

B. X and Z neurons as soft OR of Y cases

Consider a motor neuron i in the motor area Z. Whenever the neuron i is supervised to fire at value 1 at time t, a neuron j in area Y has the highest value 1. Then the weight that links these two neurons, j in area Y and i in area Z, is strengthened. Therefore, the more often neuron j fires conditioned on that neuron i fires, the higher the weight from j to i. Therefore, all the neurons in area Y that have co-fired with neuron i in area Z have non-zero weights. Therefore, either of them may cause the motor neuron:

Any of connected Y neurons fires $\Rightarrow$ Motor neuron i fires.

This soft OR relationship is due to (1) that $\mathbf{y}$ in area Y is sparse at any time and (2) multiple cases of $\mathbf{y}$ vectors fit the same neuron in area Z.

It is worth noting that the motor area does not use top-k competition since any number of motor neurons can be supervised to fire at any time. Similarly, multiple "pixel" neurons can fire concurrently in X. In other words, since X and Z are exposed to external environment, there is no guarantee that their responses are sparse. In contract, the top-k mechanism for Y can always guarantee the sparseness.

C. Prediction for image and motor

The above two properties combined enable TCM to predict patterns in X and Z based on temporal context. That is, the correspondence from temporal context to the desired Z output is based on case-based recall, as a brain inspired content addressable memory. The Theorem 6 in Section VII indicates that such a temporal context can be learned to be highly selective spatially and temporally, and of any temporal length.

The prediction for X reduces noise, stabilizes images, and suppresses unattended regions; and the prediction for Z generates learned external behaviors as outputs from Z and abstract states as input from Z.

D. Almost no local minima

Intuitively, as long as there are a sufficient number of neurons in area Y and there is a sufficient amount of training experience, the trained TCM can approximate and predict high dimensional signals in $X \times Z$ to a desired precision based on temporal context learned from past experience. This TCM scheme seems to largely avoid the problem of local minima with the existing methods, such as the error back-propagation methods [106] and other explicit nonlinear search methods [59]. This is because the TCM does not have an objective function which generates very complex rough nonlinear "terrains" through which a maximum location is sought. The generative version of TCM learns immediately and error-free as established by Theorem 1.

E. Internal sensing and actions

Weng 2007 [93] proposed a Self-Aware Self-Effecting (SASE) mental architecture, which contains internal sensors and internal effectors, in addition to external sensors and external effectors that sense and act on external world (outside the brain). With the TCM, internal actions (e.g., internal attention) involve all top-down projections from the area Z to the area Y and from the area Y to the area X but they also require other two types of connections (bottom-up and lateral) to function. External and internal sensing involve all bottom-up connections in TCM, but they also require other

types of connections (lateral and top-down) to function. The reader is referred to Weng 2007 [93] for the meaning and the importance of internal sensing and internal action.

VII. Properties

Based on the above discussion, we are ready to present major properties that are of paramount importance to temporal processing in TCM.

Let us first gain an overall perspective. The representation in the motor area of TCM is recursively used as a temporal state for subsequent cortical processing. For example, in Fig. 2, "young cat" and "kitten" should lead to the same equivalent state. If the state was not open and supervised (calibrated in general) by the teacher, there is no guarantee that internally "young cat" and "kitten" lead to the same state. If the state was not used as a top-down condition for the next internal cortical processing, there is no guarantee that all equivalent context sequences (e.g., "young cat" and "kitten") are treated exactly the same in all future processing so that the current skill is transferred to all equivalent context sequences in the future. However, due to the "skull closed" nature of autonomous development, the required internal representations inside the "skull" of TCM must emerge, and such emergence must be fully autonomous inside the skull.

A. For any AFA there is a learning TCM

In a static symbolic world, a static AFA can be handcrafted to model the complex symbolic decision process of an agent working in the symbolic world.

Learning an FA by a network has been an extensively studied subject. Although it has been known that a feedforward network is a general approximater, this theoretical result was proved based on an existence proof. How a network can effectively approximate an FA has been of keen interest in the artificial neural network community. In addition to the neural network models for temporal processing discussed above, some studies have investigated how a network can approximate an FA.

All the existing models on simulating an FA by a traditional neural network (TNN) require a handcrafted encoding of every state $q \in Q$ and a handcrafted encoding of every input symbol $\sigma \in \Sigma$. In other words, the states of the TNN cannot arbitrarily emerge like the Z area of DN. DN allows a naturally emergent representation in Z and Σ because of each internal neuron in Y learns the pattern of every $\sigma \in \Sigma$ and every $q \in Q$.

Frasconi et al. 1995 [26] used a feed-forward network to explicitly compute the state transition function $\delta : Q \times \Sigma \mapsto Q$ of an FA. Their network requires (1) a special canonical binary coding of the states so that the Hamming distance is 1 between any source state q and any target state q', (2) an additional intermediate state is added if the source state q and target state q' are the same, (3) the entire state transition function δ is known *a priori* so that their algorithm can directly compute all the weights as a batch (i.e., compiled, instead of learned incrementally). This compiled network uses a layer of logic-AND nodes followed by a layer of logic-OR nodes. Frasconi et al. 1996 proposed a radial basis function as an alternative *compiled* feed-forward network for the above logic network [27] since a finite number of samples is sufficient for completely characterizing the FA due to its symbolic nature. Omlin & Giles 1996 [71] proposed a second-order network for computing the state transition function of a fully given FA. By 2nd order, the neuronal input contains the sum of weighted multiplications (hence the 2nd order) between individual state nodes and individual input nodes. The multiplication in a 2nd order network serves as a logic AND between the state and the input symbol in the required encoding scheme. The network Omlin & Giles 1996 is also statically "programmed" by a human programmer based on a fully given FA. Forcada & Carrasco 2001 [25] gave a good survey of the related work.

The above studies aimed to successfully compute the state transition function using a programmed network from a statically given FA, but they do not generate emergent representations, do not learn, do not deal with natural input images, and do not deal with natural motor images, let alone incremental learning. In our text-based experiments discussed in Sec. VIII-B, we used an encoding for $\sigma \in \Sigma$ and $q \in Q$ but the DN works for any naturally emergent $\sigma \in \Sigma$ and $q \in Q$.

Obviously, the FA is large if it represents all the knowledge that a human has learned in his life. It seems impractical for the teacher to handcraft such an overly large FA. During autonomous development, the TCM should incrementally learn the FA though observation of FA operations, one state transition at a time. We have the following theorem.

Theorem 1 (TCM emulates AFA): Through the observation of the operations of any AFA, a TCM incrementally learns and emulates the AFA. This TCM has $|Q|$ of Z neurons and at most $|\Sigma||Q|$ of Y neurons. Its Y neurons are initialized incrementally by each newly observed vector in $(\mathbf{x}, \mathbf{z})$. The TCM emulates the AFA state transition exactly (error free) and immediately after observing each AFA state transition.

Here is a sketch of the proof while the fully detailed proof is longer than appropriate for this paper and will appear elsewhere. This proof is constructive, since its corresponds to a DP algorithm that constructs the TCM in the theorem.

Proof: First, without loss of generality, X uses a canonical representation for Σ: The i-th component of $\mathbf{x}_i \in X$ represents the i-th symbol $\sigma_i \in \Sigma$. We say that $\mathbf{x}_i$ corresponds to σ_i, denoting as $\mathbf{x}_i \equiv \sigma_i$. However, any representation for X is valid for the proof, as long as every $\sigma \in \Sigma$ corresponds to a unique $\mathbf{x}_i \in X$. Similarly, also without loss of generality, use a canonical representation for Q.

The TCM is mapped from the AFA as follows: Its X area corresponds to Σ. Its Z corresponds to Q. The areas Y and Z use the top-1 firing rule.

To emulate exactly, its Y area memorizes all observed pairs (q, σ) with $q \in Q$ and $\sigma \in \Sigma$, as all possible inputs of $\delta : \Sigma \times Q \mapsto Q$. This is done incrementally. At each time frame, observing AFA as $q \stackrel{\sigma}{\rightarrow} q'$. Feed $(\mathbf{x}, \mathbf{z}) \equiv (\sigma, q)$ to the TCM. If (σ, q) is new to the Y area, indicated by $\dot{\mathbf{v}} \cdot \dot{\mathbf{p}} < 1$ for the top winner in Y, a new Y neuron j is generated which is initialized by age 0 and $\mathbf{v}_j = \dot{\mathbf{p}}$ for $\mathbf{p} = (\dot{\mathbf{x}}, \dot{\mathbf{z}})$. The Y area computes after the neural genesis, and the new Y neuron must fire as it is the perfect winner. Otherwise, the top winner $\mathbf{v}_j$ updates. Since the Y winner neuron matches the input $\mathbf{p}$ perfectly,

during its Hebbian learning the winner neuron advances its age but does not change its weight vector $\mathbf{v}_j$.

It can be proved that every winner Y neuron matches input $\dot{\mathbf{p}}$ perfectly. Therefore, every winner neuron only advances its age but never changes its weight vector once initialized.

Since the best match in Y is always perfect, the firing neuron in Y is always unique and correct. This is also true for the firing neuron in Z because each Y neuron only links to a single correct Z neuron through Hebbian learning because of the following reasons. Whenever a Z neuron i fires as the top winner in Z, supervised during training $\mathbf{z} \equiv q'$ or during testing where $\mathbf{z}$ is left free, only one pre-synaptic neuron j in Y fires and it must be correct. From Eq. (4), the connection from the Y neuron j to the Z neuron i receives a positive gain (connects if it has not).

Because the firing Y is always correct and every Y neuron only links to a single correct Z neuron, all the firing components in the Z area are always those that should fire and all components in Z that do not fire should not indeed.

From the above reasoning, we can see that when the AFA has displayed all its $|\Sigma||Q|$ state transitions, the number of Y neurons generated is exactly $|\Sigma||Q|$ and will not increase. Each AFA state transition needs to be supervised only once. The TCM learns this transition immediately without any error. The TCM does not have any error for every state transition as soon as it has been observed from the AFA and learned by TCM through its Z supervised learning. ■

From this proof, we know that the teacher should teach the motor end of the TCM using the state of the corresponding AFA. This enables the TCM to perform state-based reasoning in the sense of the AFA. Every learned transition is successfully performed by the TCM immediately before the entire AFA is learned.

For incrementally learning the AFA which as a finite $|\Sigma||Q|$, the number of Y neurons needed is finite. For a real physical world where the Y input $\mathbf{p} = (\dot{\mathbf{x}}, \dot{\mathbf{z}})$ is real sensory and motor inputs, the number of possible inputs is infinite. Therefore, for a real world, TCM has a large but a bounded number of Y neurons. As soon as Y has run out of its neurons, the resource-bounded optimality in TCM kicks in.

Interestingly, the AFA that a TCM learns resides in the physical world (rooted in its physical causality) and in the behaviors of human teachers (e.g., based on human discovery of science). It is important to note that the entire AFA does not need to complete at any time, since the TCM agent only observe one state transition at a time from the AFA. Therefore, unlike CYC project whose goal is to construct ontology (the entire collection of human common sense knowledge), the TCM does not require that such an AFA to be completed by any human group at any time. Each human teacher knows only a part of the AFA, but that is sufficient for the TCM agent to learn a single state transition at a time from a single teacher. The TCM agent also learns directly from the physical world as part of AFA, potentially leading to discovery.

One important condition from the AFA definition is that all state transitions are consistent. This is our temporal restriction in this work. As pointed out by Weng 2011 [98], the TCM allows inconsistent supervisions at the effector end and responds optimally, but this subject is beyond the scope of this work.

In the brain of the TCM agent, internal self-organization is fully autonomous. The programmer of the DP for the TCM does not need to know what subjects that the TCM will end up learning in its "life". This is very different from an AFA agent, which is static and handcrafted by the programmer who must know the task that the AFA is supposed to execute. In this sense, the DP for many possible TCMs is an automatic programmer, programming the brains (TCMs) through interactions with the physical world like the brain does.

The AFA does not have any internal representation but the TCM has. The internal representation inside TCM provides not only a capability for handling uncertainty for each neuron, but also the internal hidden states of Y area that an SN does not have. For example, the open state of each AFA does not remember what is exactly read, but the hidden states as the internal responses of Y keep a short history of input sequence (but should not include the entire sequence). SLM, ESM, and reservoir computing have hidden states. Because they do not have open states as actions, their hidden states are not rooted in open states and therefore have very different contents.

The following properties can be systematically understood by considering a teacher who properly teaches a set of actions (i.e., concepts) that corresponds to each open state — a set of values of the active concepts, e.g., where-and-how (procedure memory) in the location motor area LM (reporting location) and the type motor area TM (reporting object type, properties, type of a spatiotemporal event, etc.) of the WWN. If the motor area TM includes vocal muscles, it does report the type information (verbally saying the type).

B. Context dependent attention

Corollary 1 (Context-dependence): Given external bottom-up input $\mathbf{x}(t)$ and top-down context $\mathbf{z}(t)$, the 3-area TCM network has three classes of internal behaviors, external (E), internal (I) and mix (M), which means that the motor output $\mathbf{z}(t+2)$ is dependent on external input $\mathbf{x}(t)$ only, dependent on internal top-down context $\mathbf{z}(t)$ only, and dependent on both, respectively.

Proof: Referring to Fig. 5, from the previous proof for AFA as a special case of TCM, we can see all we need to prove is the following: For the AFA that the TCM learns to emulate, $\delta(\sigma, q)$ is designed by the teacher to be dependent on the first argument σ only, the second argument q only, and both, respectively. This design is clearly accomplishable. ■

C. Active time warping

The phenomenon that a dynamic event can proceed at different speeds at different time points is called *time warping*. It is desirable that physical events with different time warpings but the same meaning are recognized as the same type of event (e.g., in speech recognition and dynamic visual event recognition). The way of TCM to deal with time warping is *active* in the sense that it is the learned *active* behaviors of TCM that deal with time warping.

For example, the following two sequences should be recognized as the same sequence at the motor area:

```
|w|w|w|w|w|w|_|u|u|u|_|u|u|_|_|z|z|z|z|_|
|w|w|_|u|u|u|u|_|u|u|u|u|_|z|z|_|_|_|_|_|
```

where w, u, z are words and | is a delimiter of time frames, each of which corresponds to a different t.

For this property, we have the following theorem:

Theorem 2 (Active Time Warping): The motor area of TCM can be taught to carry out active time warping.

Proof: Use the proof for Corollary 1. The teacher can design the corresponding AFA that deals with active time warping. When receiving multiple consecutive inputs that should be treated the same (e.g., silence, or stop words), the AFA stays in the same state. ■

Note that this scheme will not confuse one w sequence with two w sequences separated by a space _, since the space causes the AFA to enter a new state.

D. Temporal attention

Our major goal is to interactively train TCM so that it make sequential context-dependent decisions (actions) that require spatiotemporal, attended context in a dynamic range of the past.

Theorem 3 (Context of any temporal length): A TCM can learn contexts of any finite temporal length.

Proof: Again, like the proof of Corollary 1, the teacher designs an AFA which enters a new state after receiving each attended segment of input. Thus, the AFA can use new states to memorize the context of a sequence of any finite length. ■

As we can see, many new states result in a large AFA. In practice, equivalent states should use the same (or similar) $\mathbf{z}$ vector.

In practice, some time frames that should be disregarded and other frames should be considered as context. For example, in speech recognition, silence frames and frames of stop words should be disregarded.

Theorem 4 (Context of any temporal subset): The temporal context of top-down context $\mathbf{z}_t$ of TCM can represent any subset of the bottom-up stimuli.

Proof: Drop the frames that do not belong to the subset, using the "drop" function shown in Fig. 5. Link the other frames using the "link" function shown in Fig. 5. ■

In terms of AFA, "drop" an input corresponds to a loop to the same state and "link" corresponds to a transition to another state.

Theorem 5 (Flush): The temporal context of top-down context $\mathbf{z}_t$ of TCM can forget (flush) all the past history by enter a state that represents only the last bottom-up input, depending on learned attention.

Proof: For the AFA, enter the state that corresponds to the last bottom-up input. For the TCM, use the "drop prefix" Fig. 5. ■

Combining above three theorems gives the following theorem.

Theorem 6 (Any context): The temporal context of top-down context $\mathbf{z}_t$ of TCM can represent any subset of the bottom-up stimuli of any length of the history.

Proof: Combining the above three theorems, use Theorem 5 to start at a desired frame of the history, use Theorem 4 to keep the desired subset, and use Theorem 3 to keep context of any desired length. ■

Theorem 6 implies that the TCM agent can learn to attend to any part of spatiotemporal context, a necessity for recognizing complex visual events and understanding complex languages.

E. Time duration

The time duration task is an opposite problem of time warping. The goal of the task is to count the length of time between two events a and b.

Drew & Abbot [17] suggested to translate the membrane potential of a neuron to time. Buonomano & Merzenich [9] and Karmarkar & Buonomano [48] used a locally-recurrent network with randomly distributed, excitatory and inhibitory synapses. Their simulations showed that the states of such a network, later called Liquid State Machine (LSM) [61], after the "kick off" from a, experiences a sequence of transitions through randomly generated states. It is unlikely that all such states are guaranteed different throughout the duration, which may cause a collapse of time counting. If there is no collapse, such type of random state methods can detect a temporal window of a specified length, but seems unable to deal with general temporal contexts, such as equivalence of states. The analog fading memory [17] does not explain how the brain can hold a state for long and keep it stable.

Oscillatory patterns have been observed in EEG signals recorded either from inside the brain or from electrodes glued to the scalp. Greek letters θ, α, β, δ and γ have been used to classify EEG waves falling into frequency ranges of 4-7.5Hz, 7.5-14Hz, 14-40Hz, 0.5-4Hz, above 40Hz, respectively. From our theory, those waves are emergent phenomena of mutual inhibition and excitation among cells. They are side effects when the brain tries to sort out the winner neurons. The main point is to sort out the winners, not the side effects. Our model contains such side effects using top-k competition among neurons. Such waves can be thought of the fluctuations when the brain "sorts".

Different from the above models, TCM learns to count using self-generated counting states, as its learned behavior of estimating time duration.

Theorem 7 (Time duration): The action $\mathbf{z}_t$ of TCM can represent any finite length of time between two specified events.

Proof: Without loss of generality, suppose $\mathbf{z}_t = i$ means time i, from a. The TCM starts to count using is action $\mathbf{z}_t$ as soon as it senses a, and terminates counting when it senses b. Suppose $*$ is a distracter, other than a and b. The TCM needs to keep counting time when it sees a distracter. The following shows how TCM responds.

```
x:|*|*|a|*|*|*|*|*|*|b|*|*|...
z:|_|_|_|1|2|3|4|5|6|7|8|9|_|...
```

In AFA, counting means entering a new state for each input. ■

F. Skill transfer to new sequences

The above results can be used to understand how to use state equivalence to transfer a skill to infinitely many new sequences that the system has never learned, as the Table I column 5 summarizes.

Skill transfer is a notation well studies in psychology [16] but in machine learning this notion was still novel by the time DARPA Transfer Learning Program was established in 2005. DARPA program director Daniel Oblinger 2011 [68] wrote: "Creating a formal theory of transfer remains a critical, yet difficult, direction for future work." We attempt our theory of transfer here.

Suppose that TCM has learned $q_i \xrightarrow{\alpha} q_j$, meaning that at state q_i, the input string $\alpha \in \Sigma^*$ leads to state q_j. Stimuli sequence α leading to behavior q_j, denoted as $\xrightarrow{\alpha} q_j$, is a skill, perceptual, cognitive, or behavioral, depending on the nature of α and q_j. $q_i \xrightarrow{\alpha} q_j$ means that the skill $\xrightarrow{\alpha} q_j$ is applicable to state (setting) q_i. We define skill transfer below.

Definition 3 (Skill transfer): Suppose a skill $\xrightarrow{\alpha} q_j$ is learned conditioned on a string β: $q_0 \xrightarrow{\beta} q_i \xrightarrow{\alpha} q_j$. Then, we say that the skill is immediately transferred to another string β' if $q_0 \xrightarrow{\beta'} q_i \xrightarrow{\alpha} q_i$.

We have the following theorem.

Theorem 8 (state-based transfer): A skill $\xrightarrow{\alpha} q_j$ conditioned on string β is immediately transferred to every string β' in the set $B = \{\beta' \mid q_0 \xrightarrow{\beta'} q_i\} = [\beta]$.

Proof: All these transfers for strings in B are valid because for any $\beta' \in B$, we have $q_0 \xrightarrow{\beta'} q_i$, based on the definition of B. Then we have $q_0 \xrightarrow{\beta'} q_i \xrightarrow{\alpha} q_j$. From the definition of transfer, we conclude that the skill is transferred to all strings in set B. ■

The stings in the set B can be learned in the past but B can be further expanded in the future life of the agent.

It is important to note that in SN, the states are handcrafted and thus, skill transfer is based on handcrafted definition of states. In DN, however, state equivalence requires attention (because of distractors in the real world), and so does the success of skill transfer.

Let us look at an example called New Sentences. It is related to how we can understand a new book that we have never read. Assume that the TCM agent arrives at a ready state Λ before reading, $q_0 \longrightarrow \Lambda$. Suppose that there are four word meanings, A, B, C, D. Each word meaning i has ten synonyms $\{w_{ij} \mid j = 1, 2, ..., 10\}$, $i = 1, 2, 3, 4$, Then, there are 10000 equivalent 4-word sentences in the form of $w_{1h}w_{2i}w_{3j}w_{4k}$, $h, i, j, k = 1, 2, ..., 10$. How does the TCM agent learn skills and transfer the skills?

This problem is addressed in the following way, as illustrated by the corresponding AFA in Fig. 6. In Lesson 1, learn individual words. The x-row below denotes sensory input at each time frame, while the z-row below denotes motor output (label) at the corresponding time frame.

```
x:|a1|a1|_|_|a2|a2|a2|_|a3|a3|_|a4|a4|...
z:|_ |A |A|_|_ |A |A |A|_ |A |A|_ |A |...
```

The delay in the corresponding motor output is due to the fact

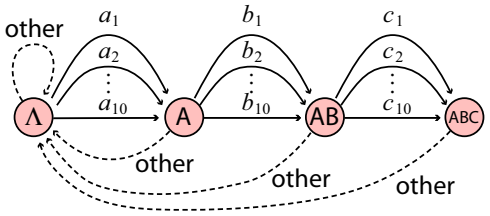


Fig. 6. The corresponding AFA for the New Sentence problem. The state "ABCD" is omitted.

that it takes two updates for the signal of sensory input to pass the area Y and appear in Z. Do the same for B, C and D. In Lesson 2, learn two-word sentences:

```
x:|a1|b1|_ |a1|b2|_ |a1|b3|_ |a1|b4|...
z:|_ |A |AB|_ |A |AB|_ |A |AB|_ |A |AB|...
```

In Lesson 3, learn 3-word sentences in a similar way. In Lesson 4, learn 4-word sentences:

```
x:|a1|b1|c1|d1 |_   |a1|b1|c1|d2 |_   |...
z:|_ |A |AB|ABC|ABCD|_ |A |AB|ABC|ABCD|...
```

The number of sentences learned in these lessons are 40, 10, 10, 10, respectively. The number of new sentences to be recognized are 0, $100 - 10$, $1000 - 10$, $10000 - 10$, respectively. Totally, the TCM learns 70 sentences, but recognizes $90 + 990 + 9990 = 11070$ new sentences. Of course, no two English words are exactly synonyms. The subtle difference is represented in area Y response, but the motor outputs are the same.

In our experiment discussed below, we used 64×64 neurons in area Y. We tested the updated TCM after every epoch through the training set. The TCM perfectly (100%) recognized all the 70 trained senses and all the 11070 new sentences from epoch 23.

Additionally, for image-based 3-D object recognition from images, the spatiotemporal network in [58] is an example of TCM. It showed that an almost perfect recognition rate has been achieve in disjoint image tests by a limited size spatiotemporal TCM. A drastic performance different between using time and not using time in object recognition from images has also been demonstrated in [58].

G. Complexity

Theorem 9 (Exponential Capacity): The number of distinguishable patterns by a cortical area with n neurons is exponential $\mathcal{O}(2^n)$.

Proof: Each neuron has least two status, firing and not firing. The number of binary firing patterns of n neurons is at least $\mathcal{O}(2^n)$. Considering $k > 1$ gives more patterns. ■

It appears that each cortical area uses sparse coding [69], [100] where relatively few neurons survive the competition to fire so that only true "expert" neurons "vote." If only k neurons are allowed to fire, the number of distinguishable patterns by a cortical area with n neurons is on the order of n^k.

This is a great advantage of distributed representation compared to a symbolic representation. While a symbolic representation potentially requires an exponential number of

symbols, no symbol is needed for these 2^n patterns using an emergent, distributed representation.

What about the state complexity that AFA faces? TCM uses emergent representations in the motor area, instead of symbolic ones. Suppose that there are n concepts and each concept has k values. The TCM has n motor subareas, each area has k neurons. This amounts to nm motor neurons, instead of the exponential k^n states with an AFA.

Theorem 10 (Complexity): The amount of computations required by a cortical area with c neurons is linear $\mathcal{O}(sc+kc)$, assuming a constant number s of average synapses per neuron and the top-k competition.

Proof: There are c neurons that need to be computed and learned. Each neuron has s synapses on average. As the computation of pre-response of each neuron requires $O(s)$ computations, a total of c neurons requires $O(sc)$ computations to generate all the pre-responses. Sorting for top-k winners for a small k/c can use the simple bubble sort algorithm, which requires a total of $O(kc)$ computations. Thus, the total computations is $\mathcal{O}(sc+kc)$. ■

With learning time t, the time complexity is $\mathcal{O}((s+k)ct)$.

The above theorem indicates that the time complexity of TCM is extremely low — linear in time, if s, k and c are constant. The number c typically depends on the complexity of the tasks to be learned and the requirements of precision. The larger the c, the more resource the area has to tessellate the observed manifolds of $X \times Z$.

Theorem 11 (Exponential AFA vs lower manifold TCM): For a task with c concepts where each concept has v values, the number of all possible symbolic states in AFA is v^c, exponential in c. The number of neurons in the motor area of TCM is vc.

Proof: To determine the number of states for the AFA, one needs to do c things. For each thing, there are v possibilities (e.g., v values of the height concept). Therefore, the number of possible states is then v^c, exponential in c. In contrast, a TCM uses c motor subareas — one motor subarea for each concept. If each concept has v values, the TCM needs a total of vc motor neurons, linear in c if v is constant. ■

This is in contrast with the brittleness of Σ and Q in the design of an AFA. The c motor areas of the corresponding TCM fire according to the experience from the environment. They are emergent, emerging from interactions with the external environments. In other words, the patterns in Z emerge automatically, without a need for a human designer to predict correctly.

Next, consider the number of transitions in the AFA and the Y neurons in TCM. In contract with AFA where the human designer of the AFA must manually select an appropriate set of transitions among the intractable underlying v^c ones, TCM automatically track the data manifolds in $(\mathbf{x}, \mathbf{z})$ through observations. The TCM seems scalable for the number of Y neurons, which could vary significantly across different species. While different symbols in an SN are simply different, the emergent representations in X and Z have natural distance defined by the pre-response value of Y — the space of inner product between normalized vectors in Eq. (2). In other words, the bounded number of Y neurons automatically interpolate across many vectors in X and Z using the inner product distance, but a symbolic representation cannot. The more Y neurons, the more details in X and Z can be predicted precisely.

VIII. Experimental Examples

As examples for the theory presented above, we discuss the results of the TCM network for two categories of inputs — video streams and text streams.

The architecture in Fig. 1 is independent to the sensory modality. For example, it is applicable to both video processing and text processing. For video streams, each frame is an image, considered as a pattern sampled from a long temporal stream. For text streams, each frame is a sample of the word that the TCM currently stares at, also considered as a pattern sampled from a long temporal stream.

Language processing typically is not treated as an issue of temporal processing, as the input unit is often considered as a logic unit of word. In contrast, we regard each word as a pattern from a temporal stream, as in Fig. 1. During natural book reading, it is possible to allow the eyes to fixate at each word for different durations of time. How does the brain knows the termination of a word? The brain perceives a pattern of multiple words at a time during reading [23]. For example, a space between two consecutive words indicates the termination of the first word. In the text processing experiment here, we used a simpler setting: We provide each word persistently for a fixed amount of time to allow the network to update twice before the next word is supplied. This is because the network needs to update twice for the input information in the sensory area X to reach the internal area Y and then the motor area Z. Each input in X consists of only a single word. The number of exposures to every word is the same (twice). The desired output is supplied at the motor end Z, always at the correct time (two-frame delay). Thus, the system does not need to sense an inter-word space to sense the termination of each word.

In all the experiments reported here, the training mode corresponds to motor-supervised learning, with which the desired action vector is imposed at motor area Z at the desired time step. Reinforcement learning of DN has been reported in [73].

A. Video processing as temporal processing

Suppose that a robot "baby" is watching an object on a rotating table while the human caregiver (teacher) interactively teaches it to sign the name of the object using its fingers, each finger being represented by a muscle neuron. As we discussed above, this is called motor-supervised learning. While the teacher lets the hand go free, the robot immediately demonstrate its performance using its fingers. The physical grounding is reflected by the realistic images from the real world and the timing between each image and the teacher supervised action. However, for precise records of performance evaluation, we chose not to use a real robot. In the real world, it is likely that the teacher makes mistakes, especially when she is tired. However, as our emphasis is on testing the simulated robot

"baby" instead of the teacher, we assume that the teacher does not make errors. This is reasonable when the object is not changed too quickly as discussed below.

In the physical world, objects come and go continuously as long as they do not move faster than the brain can update. Does top-down context assist the brain to perceive seemingly irrelevant object views as a single object [74]? On the other hand, as far as we are aware of, in pattern recognition with natural sensory inputs, there has been no report about trained systems that almost perfectly recognize a large number of unobserved natural samples[1] that are similar to, but not the same as, trained samples. Our experimental results support a positive answer for the former and created the first surprising case for the latter. That is, an almost perfect recognition rate is possible in disjoint tests if we use time.

For video processing, $\mathbf{x}(t) \in X$ represents each image frame at time t and $\mathbf{z}(t) \in Z$ represents the action that reports the cognitive supervision (input for teaching) and cognitive action (e.g., verbal output).

We used MSU-25 objects as shown in Fig. 7(a). Each object was placed on a rotary base which rotated horizontally in the full range of 360 degrees. 200 images of 56×56 pixels each were taken in sequence for each object. At the experimenter's rate of rotation, the 200 images covered about two complete rotations of $360°$. The capturing process was intentionally not too controlled, so an object varies slightly in position and size throughout its sequence. Including an additional empty (no object) class, there were $200 \times 25 + 1 = 5001$ images total. Every fifth image in each object sequence was set aside for testing. To increase the difficulty level, only gray scale images were used.

The TCM networks discussed here are fully connected: Each Y neuron is fully connected to all the neurons in X and Z. Area X, the image input (no computation), has 56×56 receptor neurons (pixels). Area Y has $20 \times 20 = 400$ neurons. Area Z, the motor area, has $26 + 1$ neurons. Thus, if the neurons in Y are considered features for many different visual views from 25 objects, each of the 400 neurons in the area needs to handle $25 \times 360/400 = 90/4$ degrees of viewing angle variability. In other words, a $90°$ of viewing angle variation is covered by **only 4 neurons** in Y. This is a task of great challenge considering the very limited neuronal resource compared with many unseen views.

The human teacher chose and taught a simple "language" for motor outputs for TCM video processing. Each motor neuron i directly outputs its pre-response value at each frame time, which indicates the confidence for recognizing object i. Therefore, during training sessions, the motor supervised vector is a binary vector, but the motor output vector during disjoint test sessions is not necessary binary. Regardless its value, the motor vector $\mathbf{z} \in Z$ in TCM is the temporal context. In this case, it represents the accumulated confidences for the types. The teacher treats the motor neuron whose pre-response value is the highest among all the motor neurons in Z to be the object class that the TCM recognizes at each time frame. As there is no guarantee that such a "winner" Z neuron does not change during the entire presentation period for each test object, the teacher treats the object class that the TCM reports most often during each object presentation period to be the object reported.

The number of neurons (k in top-k competition) allowed to fire is 15 for area Y and 8 for area Z in its testing phase. We train the networks by presenting the training sequence multiple times (epochs). The images were presented in sequence, with a few empty (no object) frames in between consecutive object sequences to mimic an object being placed in and then taken away.

A parameter α, $0 \leq \alpha < 1$, called top-down rate, used for all neurons in area Y, is the relative energy of the top-down input, and $1 - \alpha$ is that of the bottom-up input. Thus, $\alpha = 0$ corresponds to a network that does not use top-down context and $\alpha = 0.9$ indicates a network that uses a lot of top-down input. In Eq. (2), $\alpha = 0.5$.

After each epoch of training, these networks were tested using the disjoint test set (i.e., none of the tested images is in the training set), also presented in object sequences with a few empty frames in between objects. Initialized by random weights, the networks all learned fast, thanks to the double optimality of LCA discussed below. They reached 90% of the final recognition rate after the first epoch and about 99% after the second. The performance after 10 epochs is shown in Fig. 7(b). With $\alpha = 0.7$, an almost perfect recognition rate has been reached for disjoint tests.

Recently, we [96] have developed a generative version of TCM which generates Y neuron as long as the top pre-response value is less than 1. It has been proved that such a generative TCM (called Generative DN, GDN) immediately gives zero error for all the training experiences and optimal (in the sense of maximal likelihood) for new test data. That is, GDN does not need a second practice to reach the theoretically best possible performance.

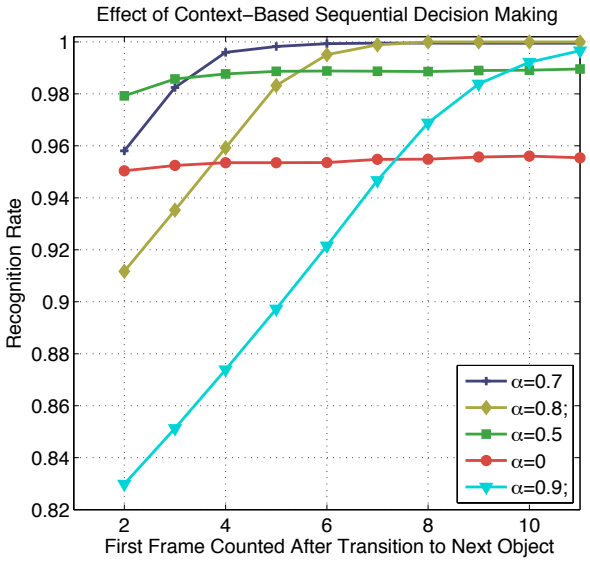
What is interesting is that at each time step the networks with top-down context generate a different top-down attention control which selects new features from the bottom-up input. As shown in Fig. 7(b), the network takes every time step to "think" with top-down attention while different views of the unknown object flow in. As shown in Fig. 7(b), the network with $\alpha = 0.7$, took an average of 5 additional views (about 200 ms if the images are updated at 30Hz) of the same object to be almost perfect in classification all the unseen views. [2] It is somewhat surprising that top-down context based "thinking" can eliminate almost all the errors in one-shot recognition (i.e., without self-generated top-down context $\alpha = 0$).

As it takes at least two time steps (frames) for the information from each image to go through the Y area and reach the motor output, the first testable frame is the 2nd frame. The plot shows that without top-down context $\alpha = 0$, there is an over 4% error. When top-down feedback takes about 70% energy ($\alpha = 0.7$), the recognition is almost perfect after the seventh sequential decision (internal attention). With more top-down energy, an almost perfect recognition rate also arrived,

[1] Not those synthesized from training samples [43].

[2] If the brain updates at 1k Hz and a human needs 100 ms to produce an action, the brain took 100 time steps.

(a)



(b)

Fig. 7. Sequential attentive thinking makes recognition almost perfect. (a) Some sample images of MSU-25 3-D objects plus a background used for training and testing. (b) Self-generated top-down context during testing makes the recognition **almost perfect** for unseen views. The vertical axis indicates the average recognition rate for unseen views by the trained limited-size network, averaged over all the test frames from the n-th frame. The horizontal axis indicates the frame number n, the frame counted after the input image stream transits to the next object. $n = 2$ is the earliest time for the information of a new image to reach the output area.

but later.

The results indicate that $\alpha = 0.8$ or larger requires relatively more views to reach an almost perfect recognition rate because the injected momentum of top-down context is larger (too "subjective" when it thinks); yet $\alpha = 0.5$ or lower does not inject a sufficient amount of top-down context to enable an almost perfect recognition (not sufficiently "subjective" when it thinks). This is the **first time**, where disjoint tests reach almost perfect recognition by a network .

B. Text processing as temporal processing

Suppose that a caregiver teaches a robot "baby" to read sentences using "word cards". One single-word card is shown to the robot at a time, while the teacher uses motor supervised mode to raise one of its fingers as output. Each finger corresponds to a meaning. The grounding is reflected by the timing between the card and the teacher's motor supervision. Again, we assume that the teacher does not make errors. For precise records of performance evaluation, we chose not to use a real robot.

To concentrate on temporal processing instead of spatial visual recognition, in simulations we used canonical representations for "word card" images and motor states — one pixel representing a different word card, and one motor neuron representing a finger. More sophisticated natural language production, where each action is represented by multiple time frames, is one of the future research goals.

In general, when real images of "word cards" are used, a very large number of "word cards" can be represented in an image of $m \times n$ pixels, with m and n fixed. Likewise, when multiple motor neurons are allowed to fire to use action pattern to show states, the number of states is exponential in the number of motor neurons. Our theory and algorithm are for such general cases. Thus, $\mathbf{x}(t) \in X$ represents a word (represented as a vector) at time t and $\mathbf{z}(t) \in Z$ represents the action input-output — action supervision for teaching or "verbal" action.

As we discussed above, emergent representations in TCM do not use explicitly hand-craft semantics (symbols). Furthermore, semantics and syntax are not separable in the TCM theory. The TCM theory regards that semantics and syntax are manifested in the actions. This seems consistent with the process of earlier language acquisition by children [74], [92], [39] who could not tell clearly which is semantics and syntax. Our theory gives a brain-like approach to understanding languages. Our theory does not consider a language to be fundamentally different from other sensorimotor skills. This seems reasonable, as human languages have many forms: visual (e.g., American Sign Language), spoken, written, and braille.

In the experiments, the human teacher supervises the TCM's actions as states, appropriate for abstraction and skill transfer. The equivalence of such states are essential for the demonstrated TCM performances.

To show the effectiveness and efficiency of the proposed TCM, the following four tasks were learnt and evaluated. All the experiments took the size of input area X as the number of words. The hidden area Y had 64×64 neurons, except for the task B which is 10×10. Neurons in the 3×3 neighborhoods around the winning neuron were updated to generate smooth neuronal areas. The size of the output area Z was a 1-D array and equals to the number of desired outputs, which is dependent on the task being taught.

Task A: New Sentences This is the task in Problem 1. The area Z is taught with the equivalent state of the corresponding AFA in Fig, 6. Experimentally, we compared TCM with MILN which classifies inputs but does not deal with time [102], [55], [58], [45]. The performances TCM are shown in Fig 8. From

the results, one can see that without the temporal context, MILN cannot obtain correct outputs for all the 2-word, 3-word and 4-word sentences. However, TCM achieved **100%** classification accuracy[3] from epoch 23 (i.e., 23 practices), including 30 learned multi-word sentences and **all 11070 new multi-word sentences**. This is a dramatic demonstration that many new subsequences and new sequences (that have not been learned) have been perfectly mapped to the correct motor actions — the power of the recursive abstract (i.e., many to one) states.

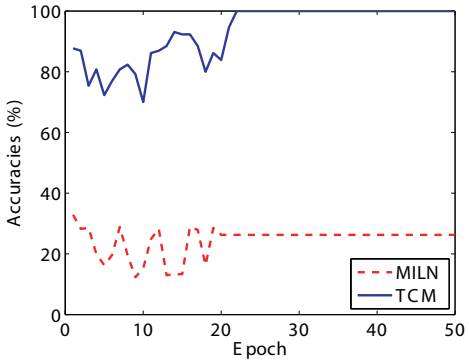


Fig. 8. Motor output accuracy versus the number of training epochs for Task A: New Sentences. All the 11070 new sentences were correctly recognized by the TCM from epoch 23.

Task B: Word sense disambiguation (WSD) This is a task for identifying which sense of a word is used in a particular context. Usually, the context (such as neighboring words) provides an evidence for disambiguation. Without context information, the WSD task could hardly achieve an acceptable performance. For example, "Apple" represents a company name or a kind of fruit. In order to show the ability of TCM to solve this task, we build a corpus with a number of logic words. Word "a_1", "a_2" and "a_3" have one sense "A". Word "b_1" has two senses "B" and "AB". When word b_1 follows word "a_1", "a_2" or "a_3", its sense is "AB". Words "b_2" and "b_3" are synonyms of "b_1". Word "c_1" also has two senses "C" and "ABC". When "c_1" follows the two words "AB", the sense becomes "ABC". Based on this task, we built a training corpus which contains 24 instances, such as $a_1 \rightarrow A$, $a_1 b_1 \rightarrow AB$, $a_1 b_2 c_2 \rightarrow ABC$, and so on. All the other combinations (e.g. $a_2 b_1 c_1$, $a_1 b_2 c_3$) are used as test data.

In the experiment, we trained the network with the 9 distinctive words and 5 distinctive meaning outputs as the states of the corresponding AFA. The result is shown in Fig. 9. From the figure one can see that the recognition accuracies provided by TCM is much better than those by MILN that does not consider time. In fact, TCM reached 100% recognition rate after epoch 34.

Fig. 10 shows the response of areas Y and Z in the two training stages of TCM, before contexts of multiple words have been learned and after, respectively. As shown in the figure, when an ambiguous word (b_1 or c_1) was received, the output at the motor area Z depends on the context so the sense of the word is disambiguated based on the word context.

[3]The brain updates very fast, about 1kHz, and it also needs reviews.

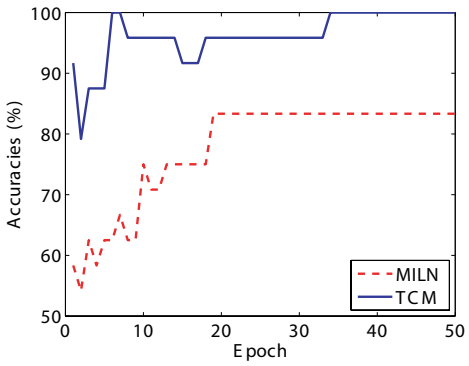


Fig. 9. Motor output accuracy versus the number of epochs through the training set for Task B: Word Sense Disambiguation.

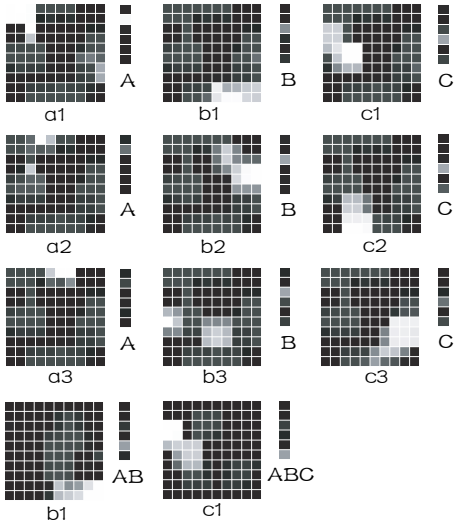


Fig. 10. The responses of area Y (shown as 10×10 images) and area Z (shown as 6-D column vectors), when the corresponding word was presented (marked below each block). The first three rows: the responses of TCM trained with single synonyms without contexts of multiple words. The last row: the responses of TCM trained with contexts of multiple words.

Task C: Part-of-speech tagging. Part-of-speech (POS) tagging, a task in natural language processing, is the process of assigning the words in a sentence to the corresponding part of speech. This is an ambiguous task without temporal context, since the same word can have two different tags in two different sentences. For example:

1) This **book** collects images of cat.
2) The students **book** the tickets from this Web site.

In the first sentence, "book" is a noun, the subject of the sentence, while in the second sentence, "book" is a verb, whose subject is "the students". The correct POS tag of a word should be determined by its context.

In this experiment, we incrementally trained a TCM network to classify words into 40 POS tags as action outputs from Z according to the Penn Treebank Tag set. The corpus we

used was extracted from the CoNLL-2000 shared task[4], which contains 8711 sentences and 211,727 tokens. Tagged from the Wall Street Journal corpus (WSJ), each token has been labeled POS tag and chunk tag. Limited by memory and time, the first 100 sentences were used for training and test, containing a total of 2,446 instances of words and 921 distinctive words. Using the canonical representation, the size of the output area Z is 1×40 (40 POS tags).

In the training phase, the network received the stream of text, with a few space characters between every two consecutive sentences. The TCM obtained drastically better training accuracies than MILN and reached 99.6% from the 21st epoch while MILN has reached only about 81%. This shows the effectiveness of processing temporal context in TCM for a large data set.

Task D: Chunking. The goal of chunking is to group sequences of words together and classify them by syntactic labels as action outputs from the area Z. Various NLP tasks can be seen as a chunking task, such as English base noun phrase identification (base NP chunking), English base phrase identification, and so on. The chunk tag of the current token is mainly determined according to the context. For example, the sentence *Mr. Carlucci served as the defense secretary in the Reagan administration.* can be divided into:

[NP Mr. Carlucci] [VP served] [PP as] [NP the defense secretary] [PP in] [NP the Reagan administration]. The chunk tags are composed of the name of the chunk type and position tag, e.g., B-NP for the first word of the noun phrase words and I-VP for each of the other words in the verb phrase words. The O chunk tag represents tokens which do not belong to any chunk. The corpus above contains eight phrase types, such as noun phrase, verb phrase and so on. Including the O tag, there are a total of $8 \times 2 + 1 = 19$ chunk tags. Thus the area Z has 19 neurons. The above example is converted into the following training format:

$Mr.$	$B-NP$
$Carlucci$	$I-NP$
$served$	$B-VP$
as	$B-PP$
$defense$	$B-NP$
$secretary$	$I-NP$
in	$B-PP$
the	$B-NP$
$Reagan$	$I-NP$
$administration$	$I-NP$
.	O

In the experiment, we trained the network with the top 100 sentences in the corpus used in Task C. Using the canonical representation, the input area X has 921 dimensions.

The TCM reached 95.2% accuracy after epoch 18. This task also shows that using temporal context can significantly increase the prediction accuracy. Since the outputs of some words are intrinsically ambiguous even with the temporal context, the accuracy cannot reach 100%. In contrast, the MINL has only reached about 86% accuracy.

Although the temporal scan is from time t to $t+1$, always advancing in time, the TCM also allow back-scan of text like a human does during normal reading if the TCM is connected with a pan-tilt head. Of course, scanning text this way requires the agent to learn more sophisticated pan-tilt behaviors and to relate the direction of the head with the text being read. In the experiments here, the agents do not have an active pan-tilt head to actively scan the text, making its "reading" simpler. We expect that in the future, a developmental robot using DN will be able to learn autonomous book reading using its pan-tilt head.

In the above two examples, although each action from TCM affects the next TCM internal operations, each action does not directly alter the next sensory inputs, which is the case with, e.g., visual navigation using a general-purpose regressor [111]. This dependency is related to the complexity of tasks that TCM learns, not a necessary condition for embodiment nor grounding, as we defined earlier. TCM allows such a dependency as one of many other possible task properties since its DP is not task specific. As we reviewed earlier, many existing methods require the human designer to hand-craft such a task-specific dependency into a learning program but TCM does not due to its task nonspecific nature.

[4]The corpus is available at http://www.cnts.ua.ac.be/conll2000/chunking/

IX. Conclusions and Discussions

The theory introduced here has proposed a set of new temporal mechanisms for both the brain-mind and the machines.

AFA: TCM is the first emergent and incremental version of AFA. By emergent, we mean that TCM is a distributed AFA, as all the representations in X, Y and Z are distributed, instead of symbolic as in AFA. By incremental, we mean that TCM is further an incremental AFA, since the underlying AFA is not handcrafted at the programming time, but is incrementally enriched from experience, using fully autonomous internal self-organization. In contrast, a modification of a symbolic AFA requires manual redesign, which is error-prone and tedious if the problem size is large.

Bayesian framework: Because each of its synapse records a scaled version of the probability, conditioned on the post-synaptic neuron, TCM is further an incremental and distributed HMM (class labels as output) and MDP (actions as output). The base AFA network of HMM or MDP is handcrafted based on the given task information. Such a hand static design is not capable of dealing with complex, dynamic, and open-ended environments and tasks.

TCM: In contrast, the internal structures of TCM emerge autonomously so that the human programmer does not need to know about tasks to learn. Further, the features in each area of TCM are dually optimal in the sense of LCA, but the meaning boundaries in an HMM and DMP network are handcrafted and do not have such an optimality. The temporal context of any length and of any subset can be attended by the TCM, through "postnatal" learning. The exponential complexity, in terms of the scan window length in online sensory processing, is converted by TCM into a linear time complexity and linear space complexity.

Many prior neural networks: The emergent representations in prior recurrent networks for temporal processing are not rooted in open motor states, limiting their power of abstraction and sequential reasoning. Although it is a neural

network, TCM is not a black box and can be taught to abstract the attended temporal context at its motor end.

Learning speed: quickset possible: Unlike prior neural networks, TCM appear to learn fast from random initial weights, around 20 practices to almost reach the peak performance. As reported in Weng 2012 [99], the GDN learning is optimal in the sense of maximum likelihood: Quickest possible toward this peak performance under limited neuronal resource and under limited training experience.

Vision: In terms of video processing, this work gives the first general framework for spatiotemporal events detection through developmental visual learning. This is a major departure from the currently model-based vision methods [22], [79], [3], [32], [110].

Language: In terms of language processing, this work represents a departure from the traditional computational modeling of language processing, pioneered by Noam Chomsky [12] and others. The language grammar is not central in the processing by TCM, but interactive associative sensorimotor experience is. We argue that such association experience by a grounded body means meaning — semantics in terms of linguistics. Many existing studies have demonstrated that sensory and motor experience played a central rule in language acquisition [28], [7], [87].

Brain: The brain's internal representations are regulated by biological mechanisms that evolved through millions of years. In the Newtonian physics, space and time are two different concepts. However, Albert Einstein's general relativity revealed that the time and space are not separable after all. Our theory here predicts how:

Inside the brain, space and time are inseparable.

The framework reported here suggests that it is computationally feasible for the brain to have no static meaning "walls" between space and time — *the spatial and temporal information is dynamically mixed almost everywhere inside the brain network*. This space-time mixture scheme is consistent with the qualitative arguments of Mauk & Buonomano [62] about the apparent absence of dedicated temporal mechanisms in the brain.

The future work includes applying TCM to other sensing modalities, such as audio streams and touch streams, to investigate its cross-modality power and possible limitations. Big data training and testing implied by autonomous development are also exciting future work.

ACKNOWLEDGEMENTS

The authors like to thank the support from, and discussions with, Drs. Xiangyang Xue and Mingmin Qi during our early attempts to study temporal capabilities of Multilayer In-place Learning Networks (MILN).

REFERENCES

[1] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9:147–169, 1985.

[2] J. R. Anderson. *Rules of the Mind*. Lawrence Erlbaum, Mahwah, New Jersey, 1993.

[3] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Proc. Computer Vision and Pattern Recognition*, pages +1–8, Miami, FL, USA, June 20 - 25, 2009.

[4] E. A. Bates, J. L. Elman, M. H. Johnson, A. Karmiloff-Smith, D. Parisi, and K. Plunkett. *Innateness and Emergentism: A Companion to Cognitive Science*. Basil Blackwell, Oxford, 1998.

[5] G. Bi and M. Poo. Synaptic modification by correlated activity: Hebb's postulate revisited. *Annual Review of Neuroscience*, 24:139–166, 2001.

[6] N. P. Bichot, A. F. Rossi, and R. Desimone. Parallel and serial neural mechanisms for visual search in macaque area v4. *Science*, 308:529–534, 2006.

[7] J. Bonvillian, K. Nelson, and V. Charrow. Language and language related skills in deaf and hearing children. *Sign Language Studies*, 12:211–250, 1976.

[8] A. Brueckner. Brains in a vat. *Journal of Philosophy*, 83(3):148–167, 1986.

[9] D. V. Buonomano and M. M. Merzenich. Temporal information transformed into a spatial code by a neural network with realistic properties. *Science*, 267:1028–1030, 1995.

[10] E. M. Callaway. Local circuits in primary visual cortex of the macaque monkey. *Annual Review of Neuroscience*, 21:47–74, 1998.

[11] E. M. Callaway. Feedforward, feedback and inhibitory connections in primate visual cortex. *Neural Networks*, 17:625–632, 2004.

[12] N. Chomsky. *Rules and Representation*. Columbia University Press, New York, 1978.

[13] J. Daly, J. Brown, and J. Weng. Neuromorphic motivated systems. In *Proc. Int'l Joint Conference on Neural Networks*, pages 2917–2914, San Jose, CA, July 31 - August 5, 2011.

[14] Y. Dan and M. Poo. Spike timing-dependent plasticity: From synapses to perception. *Physiological Review*, 86:1033–1048, 2006.

[15] G. Deco and E. T. Rolls. A neurodynamical cortical model of visual attention and invariant object recognition. *Vision Research*, 40:2845–2859, 2004.

[16] M. Domjan. *The Principles of Learning and Behavior*. Brooks/Cole, Belmont, California, fourth edition, 1998.

[17] P. J. Drew and L. F. Abbott. Extending the effects of spike-timing-dependent plasticity to behavioral timescales. *Proc. of the National Academy of Sciences of the USA*, 103(23):8876–8881, 2006.

[18] J. L. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990.

[19] J. L. Elman, E. A. Bates, M. H. Johnson, A. Karmiloff-Smith, D. Parisi, and K. Plunkett. *Rethinking Innateness: A connectionist perspective on development*. MIT Press, Cambridge, Massachusetts, 1997.

[20] A. Emami and F. Jelinek. A neural syntactic language model. *Machine Learning*, 60:195–227, 2005.

[21] S. E. Fahlman and C. Lebiere. The cascade-correlation learning architecture. Technical Report CMU-CS-90-100, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, Feb. 1990.

[22] L. Fei-Fei. One-shot learning of object categories. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

[23] D. Feitelson. *Facts and fads in beginning reading: A cross-language perspective*. Ablex, Norwood, NJ, 1988.

[24] D. J. Felleman and D. C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1:1–47, 1991.

[25] M. L. Forcada and R. C. Carrasco. Simple stable encodings of finite-state machines in dynamic recurrent networks. In J. F. Kolen and S. C. Kremer, editors, *A Field Guide to Dynamical Recurrent Networks*, pages 103–127. IEEE Press, New York, 2001.

[26] P. Frasconi, M. Gori, M. Maggini, and G. Soda. Unified integration of explicit knowledge and learning by example in recurrent networks. *IEEE Trans. on Knowledge and Data Engineering*, 7(2):340–346, 1995.

[27] P. Frasconi, M. Gori, M. Maggini, and G. Soda. Representation of finite state automata in recurrent radial basis function networks. *Machine Learning*, 23:532, 2006.

[28] R. A. Gardner and B. T. Gardner. Teaching sign language to a chimpanzee. *Science*, 165:664–672, 1969.

[29] D. George and J. Hawkins. Towards a mathematical theory of cortical micro-circuits. *PLoS Computational Biology*, 5(10):1–26, 2009.

[30] S. Grossberg and R. Raizada. Contrast-sensitive perceptual grouping and object-based attention in the laminar circuits of primary visual cortex. *Vision Research*, 40:1413–1432, 2000.

[31] S. Grossberg and A. Seitz. Laminar of receptive fields, maps and columns in visual cortex: the coordinating role of the subplate. *Cerebral cortex*, 13:852–863, 2003.

[32] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 31(10):1775–1789, 2009.

[33] G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal. The wake-sleep algorithm for unsupervised neural networks. *Science*, 268:1158–1161, 1995.

[34] G. E. Hinton, S. Osindero, and Y-. W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006.

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

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

[37] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the USA*, 79(8):2554–2558, 1982.

[38] I. Ito, R. C. Ong, B. Raman, and M. Stopfer. Sparse odor representation and olfactory learning. *Nature Neuroscience*, 11(10):1177–1184, 2008.

[39] J. M. Iverson. Developing language in a developing body: the relationship between motor development and language development. *Journal of child language*, 37(2):229–261, 2010.

[40] R. B. Ivry and J. E. Schlerf. Dedicated and intrinsic models of time perception. *Trends in Cognitive Sciences*, 12(7):273–280, 2008.

[41] H. Jaeger. Adaptive nonlinear system identification with echo state networks. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, pages 593–600. MIT Press, Cambridge, MA, 2003.

[42] F. Jelinek. Self-organized language modeling for speech recognition. In A. Waibel and K. Lee, editors, *Readings in Speech Recognition*, pages 450–506. Morgan Kaufmann, San Mateo, California, 1990.

[43] R. Jenkins and A. M. Burton. 100% accuracy in automatic face recognition. *Science*, 319(5862):435, 2008.

[44] Z. Ji and J. Weng. WWN-2: A biologically inspired neural network for concurrent visual attention and recognition. In *Proc. IEEE Int'l Joint Conference on Neural Networks*, pages +1–8, Barcelona, Spain, July 18-23, 2010.

[45] Z. Ji, J. Weng, and D. Prokhorov. Where-what network 1: "Where" and "What" assist each other through top-down connections. In *Proc. IEEE Int'l Conference on Development and Learning*, pages 61–66, Monterey, CA, Aug. 9-12, 2008.

[46] M. L. Jordan. Attractor dynamics and parallelism in a connectionist sequential machine. In *Proc. the eighth annual conference of the cognitive science society*, pages 531 – 546, Hillsdale, 1986.

[47] E. R. Kandel, J. H. Schwartz, and T. M. Jessell, editors. *Principles of Neural Science*. McGraw-Hill, New York, 4th edition, 2000.

[48] U. R. Karmarkar and D. V. Buonomano. Timing in the absence of clocks: encoding time in neural network states. *Neuron*, 53(3):427–438, 2007.

[49] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, Berlin, 3rd edition, 2001.

[50] J. E. Laird, A. Newell, and P. S. Rosenbloom. Soar: An architecture for general intelligence. *Artificial Intelligence*, 33:1–64, 1987.

[51] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of IEEE*, 86(11):2278–2324, 1998.

[52] D. B. Lenat, G. Miller, and T. T. Yokoi. CYC, WordNet, and EDR: Critiques and responses. *Communications of the ACM*, 38(11):45–48, 1995.

[53] T Lin, B. G. Horne, P. Tino, and C. L. Giles. Learning long-term dependencies in narx recurrent neural networks. *IEEE Transactions on Neural Networks*, 7(6):1329–1338, 1996.

[54] W. S. Lovejoy. A survey of algorithmic methods for partially observed Markov decision processes. *Ann. Operations Research*, 28:47–66, 1991.

[55] M. Luciw and J. Weng. Topographic class grouping with applications to 3d object recognition. In *IEEE World Congress on Computational Intelligence*, pages +1–6, Hong Kong, June 1-6, 2008.

[56] M. Luciw and J. Weng. Where What Network 3: Developmental top-down attention with multiple meaningful foregrounds. In *Proc. IEEE Int'l Joint Conference on Neural Networks*, pages 4233–4240, Barcelona, Spain, July 18-23, 2010.

[57] M. Luciw and J. Weng. Where What Network 4: The effect of multiple internal areas. In *Proc. IEEE 9th Int'l Conference on Development and Learning*, pages 311–316, Ann Arbor,, August 18-21, 2010.

[58] M. Luciw, J. Weng, and S. Zeng. Motor initiated expectation through top-down connections as abstract context in a physical world. In *IEEE Int'l Conference on Development and Learning*, pages +1–6, Monterey, CA, Aug. 9-12, 2008.

[59] D. G. Luenberger. *Optimization by Vector Space Methods*. Wiley, New York, 1969.

[60] M. Lukosevicius and H. Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.

[61] W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002.

[62] M. D. Mauk and D. V. Buonomano. The neural basis of temporal processing. *Annual Review of Neuroscience*, 27:307–340, 2004.

[63] J. L. McClelland, D. E. Rumelhart, and The PDP Research Group, editors. *Parallel Distributed Processing*, volume 2. MIT Press, Cambridge, Massachusetts, 1986.

[64] R. Miikkulainen, J. A. Bednar, Y. Choe, and J. Sirosh. *Computational Maps in the Visual Cortex*. Springer, Berlin, 2005.

[65] M. Minsky. Logical versus analogical or symbolic versus connectionist or neat versus scruffy. *AI Magazine*, 12(2):34–51, 1991.

[66] Y. Munakata and J. L. McClelland. Connectionist models of development. *Developmental Science*, 6(4):413–429, 2003.

[67] E. Niebur, C. Koch, and C. Rosin. An oscillation-based model for the neural basis of attention. *Vision Research*, 33:2789–2802, 1993.

[68] D. Oblinger. Toward a computational model of transfer. *AI Magazine*, 32(2):126–128, 2011.

[69] B. A. Olshaushen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, June 13, 1996.

[70] B. A. Olshaushen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy used by v1? *Vision Research*, 37(23):3311–3325, 1997.

[71] C. W. Omlin and C. L. Giles. Constructing deterministic finite-state automata in recurrent neural networks. *Journal of the ACM*, 43(6):937–972, 1996.

[72] R. W. Paine and J. Tani. How hierarchical control self-organizes in artificial adaptive systems. *Adaptive Behavior*, 13(3):211–225, 2005.

[73] S. Paslaski, C. VanDam, and J. Weng. Modeling dopamine and serotonin systems in a visual recognition network. In *Proc. Int'l Joint Conference on Neural Networks*, pages 3016–3023, San Jose, CA, July 31 - August 5, 2011.

[74] J. Piaget. *The Construction of Reality in the Child*. Basic Books, New York, 1954.

[75] M. L. Puterman. *Markov Decision Processes*. Wiley, New York, 1994.

[76] S. Quartz and T. J. Sejnowski. The neural basis of cognitive development: A constructivist manifesto. *Behavioral and Brain Sciences*, 20(4):537–596, 1997.

[77] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of IEEE*, 77(2):257–286, 1989.

[78] L. R. Rabiner. Toward vision 2001: Voice and audio processing considerations. *AT&T Technical Journal*, 74(2):4–13, 1995.

[79] D. Ramanan. Learning to parse images of articulated bodies. In *Proc. Neural Info. Proc. Systems (NIPS)*, pages +1–8, Vancouver, Canada, Dec. 4-9, 2006.

[80] A. S. Reber, S. M. Kassin, S. Lewis, and G. Cantor. On the relationship between implicit and explicit modes in the learning of a complex rule structure. *Journal of Experimental Psychology: Human Learning and Memory*, 6(5):492–502, 1980.

[81] P. R. Roelfsema and A. van Ooyen. Attention-gated reinforcement learning of internal representations for classification. *Neural Computation*, 17:2176–2214, 2005.

[82] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group. *Parallel Distributed Processing*, volume 1. MIT Press, Cambridge, Massachusetts, 1986.

[83] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Upper Saddle River, New Jersey, 3rd edition, 2010.

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

[85] Y. F. Sit and R. Miikkulainen. Self-organization of hierarchical visual maps with feedback connections. *Neurocomputing*, 69:1309–1312, 2006.

[86] X. Song, W. Zhang, and J. Weng. Where-what network 5: Dealing with scales for objects in complex backgrounds. In *Proc. Int'l Joint*

Conference on Neural Networks, pages 2795–2802, San Jose, CA, July 31 - August 5, 2011.

[87] I. Stockman, editor. *Movement and Action in Learning and Development: Clinical Implications for Pervasive Developmental Disorders*. Elsevier Academic Press, San Diego, California, 2004.

[88] 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):59192, 2005.

[89] M. Sur and C. A. Leamey. Development and plasticity of cortical areas and networks. *Nature Reviews Neuroscience*, 2:251–262, 2001.

[90] M. Sur and J. L. R. Rubenstein. Patterning and plasticity of the cerebral cortex. *Science*, 310:805–810, 2005.

[91] R. S. Sutton, D. Precup, and S. Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112:181–211, 1999.

[92] L. S. Vygotsky. *Thought and language*. MIT Press, Cambridge, Massachussetts, 1962. trans. E. Hanfmann & G. Vakar.

[93] J. Weng. On developmental mental architectures. *Neurocomputing*, 70(13-15):2303–2323, 2007.

[94] J. Weng. A 5-chunk developmental brain-mind network model for multiple events in complex backgrounds. In *Proc. Int'l Joint Conf. Neural Networks*, pages 1–8, Barcelona, Spain, July 18-23, 2010.

[95] J. Weng. A general purpose brain model for developmental robots: The spatial brain for any temporal lengths. In *Proc. Workshop on Bio-Inspired Self-Organizing Robotic Systems, IEEE Int'l Conference on Robotics and Automation*, pages +1–6, Anchorage, Alaska, May 3-8, 2010.

[96] J. Weng. Three theorems about developmental networks and the proofs. Technical Report MSU-CSE-11-9, Department of Computer Science, Michigan State University, East Lansing, Michigan, May 12, 2011.

[97] J. Weng. Three theorems: Brain-like networks logically reason and optimally generalize. In *Proc. Int'l Joint Conference on Neural Networks*, pages 2983–2990, San Jose, CA, July 31 - August 5, 2011.

[98] J. Weng. Why have we passed "neural networks do not abstract well"? *Natural Intelligence: the INNS Magazine*, 1(1):13–22, 2011.

[99] J. Weng. Symbolic models and emergent models: A review. *IEEE Trans. Autonomous Mental Development*, 4(1):29–53, 2012.

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

[101] J. Weng and M. D. Luciw. Optimal in-place self-organization for cortical development: Limited cells, sparse coding and cortical topography. In *Proc. 5th Int'l Conference on Development and Learning (ICDL'06)*, pages +1–7, Bloomington, IN, May 31 - June 3, 2006.

[102] J. Weng, T. Luwang, H. Lu, and X. Xue. Multilayer in-place learning networks for modeling functional layers in the laminar cortex. *Neural Networks*, 21:150–159, 2008.

[103] 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.

[104] J. Weng, Y. Shen, M. Chi, and X. Xue. Temporal context as cortical spatial codes. In *Proc. Int'l Joint Conference on Neural Networks*, Atlanta, Georgia, June 14-19, 2009.

[105] J. Weng, Q. Zhang, M. Chi, and X. Xue. Complex text processing by the temporal context machines. In *Proc. IEEE 8th Int'l Conference on Development and Learning*, pages +1–8, Shanghai, China, June 4-7, 2009.

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

[107] J. C. Wiemer. The time-organized map algorithm: Extending the self-organizing map to spatiotemporal signals. *Neural Computation*, 15:1143–1171, 2003.

[108] A. K. Wiser and E. M. Callaway. Contributions of individual layer 6 pyramidal neurons to local circuitry in macaque primary visual cortex. *Journal of neuroscience*, 16:2724–2739, 1996.

[109] Y. Yamashita and J. Tani. Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment. *PLoS Computational Biology*, 4(11):e1000220, 2008.

[110] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *Proc. Computer Vision and Pattern Recognition*, pages +1–8, San Francisco, CA, June 15-17, 2010.

[111] S. Zeng and J. Weng. Online-learning and attention-based approach to obstacle avoidance using a range finder. *Journal of Intelligent and Robotic Systems*, 50(3):219–239, 2007.

Juyang Weng (S85-M88-SM05-F09) received the BS degree in computer science from Fudan University, Shanghai, China, in 1982, and M. Sc. and PhD degrees in computer science from the University of Illinois at Urbana-Champaign, in 1985 and 1989, respectively.

He is currently a professor of Computer Science and Engineering at Michigan State University, East Lansing. He is also a faculty member of the Cognitive Science Program and the Neuroscience Program at Michigan State University. Since the work of Cresceptron (ICCV 1993), he expanded his research interests in biologically inspired systems, especially the autonomous development of a variety of mental capabilities by robots and animals, including perception, cognition, behaviors, motivation, and abstract reasoning skills. He has published over 250 research articles on related subjects, including task muddiness, intelligence metrics, mental architectures, vision, audition, touch, attention, recognition, autonomous navigation, natural language understanding, and other emergent behaviors. He coauthored with T. S. Huang and N. Ahuja a research monograph titled *Motion and Structure from Image Sequences* and authored a book titled *Natural and Artificial Intelligence: Computational Introduction to Computational Brain-Mind*.

Dr. Weng is an Editor-in-Chief of the *International Journal of Humanoid Robotics*, the Editor-in-Chief of the *Brain-Mind Magazine*, and an associate editor of the *IEEE Transactions on Autonomous Mental Development*. He was a Program Chairman of the NSF/DARPA funded Workshop on Development and Learning 2000 (1st ICDL), a Program Chairman of the 2nd ICDL (2002), the chairman of the Autonomous Mental Development Technical Committee of the IEEE Computational Intelligence Society (2004-2005), the Chairman of the Governing Board of the International Conferences on Development and Learning (ICDLs) (2005-2007), a General Chairman of 7th ICDL (2008), the General Chairman of 8th ICDL (2009), an associate editor of the *IEEE Transactions on Pattern Recognition and Machine Intelligence*, and an associate editor of the *IEEE Transactions on Image Processing*.

Matthew Luciw received the M.S. and Ph.D. degrees in Computer Science from Michigan State University in 2006 and 2010, respectively. He is currently a researcher at the Swiss AI lab, IDSIA (Istituto Dalle Molle di Studi sull'Intelligenza Artificiale). His research interests include Autonomous Mental Development, Unsupervised Learning, Reinforcement Learning, Neural Networks, and Artificial Curiosity. His web page: www.idsia.ch/~luciw. He is a member of the IEEE.

Qi Zhang is an associate professor in the School of Computer Science, Fudan University. He received his undergraduate degree in Computer Science and Technology, Shandong University in 2003. His Dr. degree in Computer Science was received from Fudan University, in 2009. From January 2005 to January 2006, he served as a Research Intern at the Bosch Research and Technology Center, Palo Alto, USA. Since April 2009, he has been with the School of Computer Science, Fudan University. His main research interests include natural language processing, and information retrieval.