Neural Modeling of Episodic Memory: Encoding, Retrieval, and Forgetting

Wenwen Wang, Budhitama Subagdja, Ah-Hwee Tan, *Senior Member, IEEE*, and Janusz A. Starzyk, *Senior Member, IEEE*

***Abstract*— This paper presents a neural model that learns episodic traces in response to a continuous stream of sensory input and feedback received from the environment. The proposed model, based on fusion adaptive resonance theory (ART) network, extracts key events and encodes spatio-temporal relations between events by creating cognitive nodes dynamically. The model further incorporates a novel memory search procedure, which performs a continuous parallel search of stored episodic traces. Combined with a mechanism of gradual forgetting, the model is able to achieve a high level of memory performance and robustness, while controlling memory consumption over time. We present experimental studies, where the proposed episodic memory model is evaluated based on the memory consumption for encoding events and episodes as well as recall accuracy using partial and erroneous cues. Our experimental results show that: 1) the model produces highly robust performance in encoding and recalling events and episodes even with incomplete and noisy cues; 2) the model provides enhanced performance in a noisy environment due to the process of forgetting; and 3) compared with prior models of spatio-temporal memory, our model shows a higher tolerance toward noise and errors in the retrieval cues.**

***Index Terms*— Adaptive resonance theory-based network, agent, episodic memory, forgetting, hierarchical structure, memory robustness, Unreal Tournament.**

I. Introduction

EPISODIC memory is a special class of memory system that allows one to remember his/her own experiences in an explicit and conscious manner [1]. Although episodic memory is considered to be less important than semantic memory, recent research has found episodic memory to be crucial in supporting many cognitive capabilities, including concept formation, representation of events in spatio-temporal dimension, and record of progress in goal processing [2]. Additional evidences from cognitive neuroscience also imply its importance during learning about context and about configurations of stimuli. In particular, Morgan and Squire have shown that during reinforcement learning tasks, hippocampus (an area of the brain believed to be the place of episodic memory) is critical for representing relationships between stimuli independent of their associations with reinforcement [3]. The specific functionalities mentioned above suggest that episodic memory should not be just a storage of one's past experiences, but should support the representation of complex conceptual and spatio-temporal relations among one's experienced events and situations. Many existing computational models of episodic memory have been capable of encoding events and relations between events (e.g., [4] and [5]). However, most still have limitations in capturing complex concepts and situations. On the other hand, those models supporting the intricate relations of concepts and events are not able to process complex sequences of events (e.g., [6]–[8]).

In this paper, we present a computational model called electromagnetic adaptive resonance theory (EM-ART) for encoding of episodic memory in terms of events as well as spatio-temporal relations between events. The model can be incorporated into an autonomous agent for encoding its individual experience, which can be retrieved later for reasoning in real time. Based on a generalization of fusion adaptive resonance theory (ART) [9], the EM-ART model supports event encoding in the form of multiple-modal patterns. An episodic encoding scheme is introduced that allows temporal sequences of events to be learned and recognized. The model further incorporates a novel memory search procedure, which performs parallel search of stored episodic traces continuously in response to potentially imperfect search cues. Extending from our previous work presented in [10], we further enhance EM-ART with a mechanism of gradual forgetting. The forgetting mechanism removes unimportant and outdated events from the episodic memory and enables the model to maintain a manageable level of memory consumption over a possibly infinite time period. This is a feature crucially required by real-time systems.

We have conducted experimental studies on the proposed model through two different applications. The first application is a word recognition task, wherein the proposed model is used to learn a set of words. The performance is measured by the accuracies of retrieving the learned words given their noisy versions. Compared with existing models of spatio-temporal memory, the experiment results show that the EM-ART model is one of the best models in terms of retrieval performance. We also evaluate EM-ART in a first person shooting game called Unreal Tournament, wherein EM-ART is used to learn episodic memory based on an agent's encounters in the game. Experiments show that EM-ART produces a robust level of performance in encoding and recalling events and episodes using various types

Manuscript received August 23, 2011; revised June 29, 2012; accepted July 1, 2012. Date of publication August 6, 2012; date of current version September 10, 2012. This work was supported in part by DSO National Laboratories under Research Grant DSOCL11258.

W. Wang, B. Subagdja, and A.-H. Tan are with the School of Computer Engineering, Nanyang Technological University, 639798 Singapore (e-mail: wa0003en@ntu.edu.sg; budhitama@ntu.edu.sg; asahtan@ntu.edu.sg).

J. A. Starzyk is with the School of Electrical Engineering and Computer Science, Russ College of Engineering and Technology, Ohio University, Athens, OH 45701-2979 USA, and also with the University of Information Technology and Management, Rzeszów 35-959, Poland (e-mail: starzykj@ohiou.edu).

Digital Object Identifier 10.1109/TNNLS.2012.2208477

of input queries, involving incomplete and noisy cues. This is in comparison with the long term memory (LTM) model, which is another best performing model in the word recognition task. By further examining the effects of forgetting, we find that the incorporated forgetting mechanism also promotes more efficient and robust learning by continuously pruning erroneous and outdated patterns.

The rest of this paper is organized as follows. Section II discusses the issues and challenges. Section III presents the architecture of our proposed episodic memory model. Sections IV and V present the algorithms and processes for event and episode encoding and retrieval, respectively. Section VI discusses the forgetting mechanism incorporated in the proposed model. Sections VII and VIII investigate the performance and robustness of the proposed model in the word recognition task and the shooting game, respectively. Section IX provides a brief discussion and comparison of selected work on episodic memory models. The final section concludes and highlights future work.

II. Issues and Challenges

A. Memory Formation

As discussed in Section I, two basic elements of episodic memory are events and episodes. An event can be described as a snapshot of experience. Usually, by aggregating attributes of interest, a remembered event can be used to answer critical questions about the corresponding experience, such as what, where, and when. On the other hand, an episode can be considered as a temporal sequence of events.

To enable efficient encoding of events and episodes, an episodic memory model should be able to distinguish between distinct events and episodes with a well-defined matching scheme. The basic challenge regarding building the memory storage matching scheme is, on one hand, the novelty detection should be sufficiently strict to distinguish highly similar but semantically different events (e.g., "Mary borrowed a book from Emma yesterday" is different from "Mary borrowed a book from Bob yesterday"). On the other hand, it should also be loose enough to tolerate minor differences for events within a single episode, such as slight changes within observed events and their temporal order. Hence, the critical characteristic for the matching scheme is its high efficiency in determining the significant differences while tolerating all minor variances for both events and episodes encoding. Therefore, an efficient matching scheme should also lead to a parsimonious memory storage as well as faster memory operations.

B. Memory Retrieval

We identify three major tasks in episodic memory retrieval, namely event detection, episode recognition, and episode recall, described as follows.

1) Event detection refers to the recognition of a previously learned event based on a possibly incomplete description of the current situation. The episodic memory model should be able to search for similar memorized events, which can be used to complete or refine the given description.
2) Episode recognition refers to the identification of a stored episode in the episodic memory in response to a partial event sequence. Following the effect of episode recognition, episodic memory model may also perform event completion if the presented event sequence has missing parts. Two basic requirements of episode recognition include: 1) tolerance to incomplete cues, which only form parts of the stored episodes and 2) tolerance to errors, for example, noise in event attributes and variations in the order of event sequences.
3) Episode recall is the playback of episode(s) in response to an external cue, such as "what did I do yesterday? When a cue is presented, episodic memory answers the cue with the most closely matched episode according to its similarity. During the episode playback, compared with the stored information, an exemplar cue may present minor disparities in individual event representations as well as their temporal orderings. The episodic memory model should be able to identify and tolerate this imperfection during recall.

C. Forgetting

Many studies (e.g., [11] and [12]) have indicated that the memory traces in the hippocampus are not permanent and are occasionally transferred to neocortical areas in the brain through a consolidation processes. This implies that forgetting should exist in episodic memory to avoid possible information overflow. Forgetting in the episodic memory helps to preserve and strengthen important or frequently used episodes, and remove (or forget) unimportant ones. Forgetting is not only a natural and desired characteristic of biological intelligence, it is also a prevalent operation in continuous real time artificial models that gradually learn how to operate in a given environment. More importantly, it is a necessary condition for promoting efficient memory storage, as well as fast and accurate operation of episodic memory in real-time environments.

D. Summary

Taking the above into consideration, an episodic memory model should satisfy the following basic requirements: 1) efficient event representation describing complex situations and events; 2) efficient episode representation for exploring spatio-temporal relations among events which form the episode; 3) well-defined generalizations on representations, which accurately distinguish critical and irrelevant differences among them (for both events and episodes); 4) high level of tolerance to incomplete or noisy cues; 5) fast memory operations, including memory encoding and retrieving; 6) tracking the importance of events and episodes in real time based on rewards, surprises, emotions, interpretation, and access frequency; and 7) forgetting mechanism to deal with the limited memory capacity.

III. Proposed Model

Our proposed episodic memory model, called EM-ART, is built by hierarchically joining two multichannel self-organizing fusion ART neural networks. Based on ART [13],

which has led to a steadily developed family of neural learning architectures [14], [15], fusion ART dynamics offers a set of universal computational processes for encoding, recognition, and reproduction of patterns.

As shown in Fig. 1, the model consists of three layers of memory fields: F_1, F_2, and F_3. The F_1 layer is connected with the working memory to hold the activation values of all situational attributes. Based on the F_1 pattern of activations, a cognitive node in F_2 is selected and activated as a recognition of the event. Following that, the activation pattern of an incoming event can be learned by adjusting the weights in the connections between F_1 and F_2.

Besides categorizing events, the F_2 layer also acts as a medium-term memory buffer for event activations. A sequence of events produces a series of activations in F_2. The activations in F_2 decay over time such that a graded pattern of activations is formed representing the order of the sequence. This activity pattern, which represents an episode, is similarly learned as weighted connections between F_2 and the selected category in F_3.

Once an episode is recognized through a selected node in F_3, the complete episode can be reproduced by a top down activation process (readout) from F_3 to F_2. The events in the episode can also be reproduced by reading out the activations from F_2 to F_1 following the order of the sequence held in the F_2 layer.

The computational principles and algorithms used for encoding, storing, and retrieving events and episodes are described in details in the following sections.

IV. Event Encoding and Retrieval

An event consists of attributes characterizing what, where, and when an event occurs. Fig. 2 shows an example of the structure of an input event based on the Unreal Tournament domain [16]. This structure is also used in the experiments for evaluating EM-ART. In the structure shown, the location is expressed using a 3-D cartesian coordinate system; other task and internal states include the observed distance from the enemy (another agent), the availability of collectable items, and the agent's health and ammo level.

There are four behavior choices (actions) available for the agent, including running around, collecting items, escaping from battle, and engaging in fire. The consequence of a battle situation (e.g., kill and damages) is presented to the model as a reward value.

A. Fusion ART

Fusion ART network is used to learn individual events encoded as weighted connections between the F_1 and F_2 layers. In this case, an event is represented as a multichannel input vector. Fig. 3 illustrates the fusion ART architecture, which may be viewed as an ART network with multiple input fields. Each event's attribute is represented as the activity of a node in the corresponding input field.

For completeness, a summary of the fusion ART dynamics is given below.

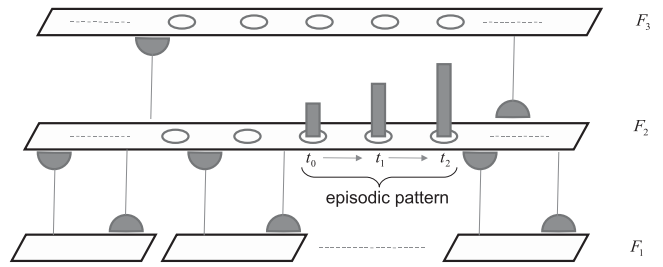

Fig. 1. Proposed neural network architecture of the episodic model.

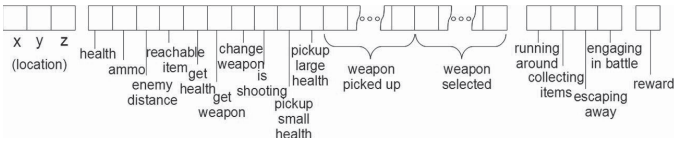

Fig. 2. Event encoding.

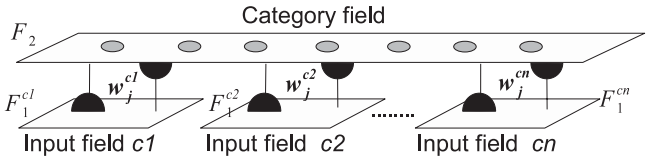

Fig. 3. Fusion ART architecture.

Input Vectors: Let $\mathbf{I}^k = (I_1^k, I_2^k, \ldots, I_n^k)$ denote an input vector, where $I_i^k \in [0, 1]$ indicates the input i to channel k, for $k = 1, \ldots, n$. With complement coding, the input vector $\mathbf{I}^k$ is augmented with a complement vector $\bar{\mathbf{I}}^k$ such that $\bar{I}_i^k = 1 - I_i^k$.

Input Fields: Let F_1^k denote an input field that holds the input pattern for channel k. Let $\mathbf{x}^k = (x_1^k, x_2^k, \ldots, x_n^k)$ be the activity vector of F_1^k receiving the input vector $\mathbf{I}^k$ (including the complement).

Category Fields: Let F_i denote a category field and $i > 1$ indicate that it is the ith field. The standard multichannel ART has only one category field, which is F_2. Let $\mathbf{y} = (y_1, y_2, \ldots, y_m)$ be the activity vector of F_2.

Weight Vectors: Let $\mathbf{w}_j^k$ denote the weight vector associated with the jth node in F_2 for learning the input pattern in F_1^k.

Parameters: Each field's dynamics is determined by choice parameters $\alpha^k \geq 0$, learning rate parameters $\beta^k \in [0, 1]$, contribution parameters $\gamma^k \in [0, 1]$, and vigilance parameters $\rho^k \in [0, 1]$.

The dynamics of a multichannel ART can be considered as a system of continuous resonance search processes comprising the basic operations as follows.

Code Activation: A node j in F_2 is activated by the choice function

$$T_j = \sum_{k=1}^{n} \gamma^k \frac{|\mathbf{x}^k \wedge \mathbf{w}_j^k|}{\alpha^k + |\mathbf{w}_j^k|} \tag{1}$$

where the fuzzy AND operation $\wedge$ is defined by $(\mathbf{p} \wedge \mathbf{q})_i \equiv \min(p_i, q_i)$, and the norm $|.|$ is defined by $|\mathbf{p}| \equiv \sum_i p_i$ for vectors $\mathbf{p}$ and $\mathbf{q}$.

Code Competition: A code competition process selects a F_2 node with the highest choice function value. The winner

is indexed at J where

$$T_J = \max\{T_j : \text{for all } F_2 \text{ node } j\}. \quad (2)$$

When a category choice is made at node J, $y_J = 1$; and $y_j = 0$ for all $j \neq J$ indicating a winner-take-all strategy.

Template Matching: A template matching process checks if resonance occurs. Specifically, for each channel k, it checks if the match function m_J^k of the chosen node J meets its vigilance criterion such that

$$m_J^k = \frac{|\mathbf{x}^k \wedge \mathbf{w}_J^k|}{|\mathbf{x}^k|} \geq \rho^k. \quad (3)$$

If any of the vigilance constraints is violated, mismatch reset occurs or T_J is set to 0 for the duration of the input presentation. Another F_2 node J is selected using choice function and code competition until a resonance is achieved. If no selected node in F_2 meets the vigilance, an uncommitted node is recruited in F_2 as a new category node.

Template Learning: Once a resonance occurs, for each channel k, the weight vector $\mathbf{w}_J^k$ is modified by the following learning rule:

$$\mathbf{w}_J^{k(\text{new})} = (1 - \beta^k)\mathbf{w}_J^{k(\text{old})} + \beta^k \left(\mathbf{x}^k \wedge \mathbf{w}_J^{k(\text{old})}\right). \quad (4)$$

Activity Readout: The chosen F_2 node J may perform a readout of its weight vectors to an input field F_1^k such that $\mathbf{x}^{k(\text{new})} = \mathbf{w}_J^k$.

A fusion ART network, consisting of different input (output) fields and a category field, is a flexible architecture that can be made for a wide variety of purposes. The neural network can learn and categorize inputs and can be made to map a category to some predefined fields by a readout process to produce the output. Another important feature of the fusion ART network is that no separate phase of operation is necessary for conducting recognition (activation) and learning. Learning can be conducted by adjusting the weighted connections while the network searches and selects the best matching node. When no existing node can be matched, a new node is allocated to represent the new pattern. Hence, the network can grow in response to novel patterns.

B. Algorithm for Event Encoding and Retrieval

Based on the above description of fusion ART, an event can be encoded as an input vector to the network such as the one shown in Fig. 2.

The recognition task can be realized by a bottom-up activation given the input vector, using the standard operations of fusion ART. On the other hand, the top-down activation (readout operation) achieves the recall task. Fig. 4 illustrates the bottom-up and top-down operations for learning, recognition, and recalling an event.

More specifically, the algorithm for learning and recognizing events can be described as Algorithm 1.

The algorithm for event recognition and encoding is designed to handle complex sequences, involving repetition of events. The iteration condition in line 3 Algorithm 1 ensures that the same node will not be selected if it has been selected previously as a matching category in the same episode. This leads to the creation of a new event category when the event pattern is repeated in a sequence (episode). One important parameter for event recognition and encoding is ρ^k, the vigilance parameter for each input channel k in F_1. The vigilance values are used as thresholds for the template matching process, as described in Section IV-A. If the same vigilance value is applied to all input channels in F_1 layer, ρ^e is introduced to represent this unified vigilance value for encoding and retrieval of events.

Algorithm 1 Event Encoding

1 Given an input pattern of event as vector $\mathbf{I}^k$ in F_1
2 Activate every node j in F_2 by choice function $T_j = \sum_{k=1}^{n} \gamma^k \frac{|\mathbf{x}^k \wedge \mathbf{w}_j^k|}{\alpha^k + |\mathbf{w}_j^k|}$
3 select node J such that $T_J = \max\{T_j : \text{for all } F_2 \text{ node } j\}$,
4 set node activation $y_J \leftarrow 1$
5 WHILE match function $m_J^k = \frac{|\mathbf{x}^k \wedge \mathbf{w}_J^k|}{|\mathbf{x}^k|} < \rho^k$ (not in resonance)
OR J was selected previously
6 deselect and reset J by $T_J \leftarrow 0$, $y_J \leftarrow 0$
7 select another node J with $T_J = \max\{T_j : \text{for all } F_2 \text{ node } j\}$
8 IF no matching (resonance) J can be found in F_2
9 THEN let $J \leftarrow J^0$, where J^0 is a newly recruited uncommitted node in F_2
10 learn J as a novel event with $\mathbf{w}_J^{k(\text{new})} = \mathbf{x}^k$

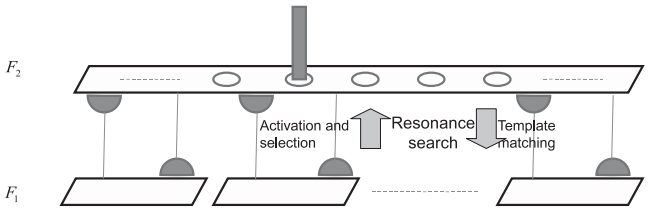

Fig. 4. Operations between F_1 and F_2.

V. Episode Learning and Retrieval

A. Episode Representation and Learning Algorithm

A crucial part of episodic memory is to encode the sequential or temporal order between events. However, in the standard model of fusion ART, this feature of sequential representation is missing. The EM model proposed in this paper extends the fusion ART model so that it can associate and group patterns across time.

Specifically, we adopt the method of invariance principle [17], [18], which suggests that activation values can be retained in a working memory (neural field) in such a way that the temporal order in which they occur are encoded by their activity patterns. To retain the temporal order, each entry of activation item multiplicatively modifies the activity of all previous items. Based on the multiplying factor, an analog pattern emerges in the neural field reflecting the order the events are presented. Thus, the temporal order of items in

a sequence, encoded as relative ratios between their values, remains invariant.

This method emulates the characteristic of serial learning conforming psychological data about human working memory [17]. The approach can be simplified by replacing multiplication with adding/subtracting operations, which is called gradient encoding, and has been successfully applied to iFALCON, a belief-desire-intention agent architecture comprised of fusion ARTs [19].

To represent a sequence in our EM model, the invariance principle is applied, so that an activation value in F_2 indicates a time point or a position in an ordered sequence. The most recently activated node in F_2 has the maximum activation of 1, while the previously selected ones are multiplied by a certain factor decaying the values over time. Suppose $t_0, t_1, t_2, \ldots, t_n$ denote the time points in an increasing order, and y_{t_i} is the activity value of the node that is activated or selected at time t_i, the activation values in F_2 form a certain pattern such that $y_{t_i} > y_{t_{i-1}} > y_{t_{i-2}} > \cdots > y_{t_{i-n}}$ holds where t_i is the current or the latest time point. This pattern of activation also possesses or exhibits the so-called recency effect in short-term memory, in which a recently presented item has a higher chance to be recalled from the memory.

The process of episode learning in EM-ART is shown in Fig. 5. While a newly activated node has an activation of 1, the activation value of any other node j in F_2 is decayed in each time step so that $y_j^{(\text{new})} = y_j^{(\text{old})}(1-\tau)$, where y_j is the activation value of the jth node in F_2 and $\tau \in (0, 1)$ is the decaying factor.

Concurrently, the sequential pattern can be stored as weighted connections in the fusion ART network. As mentioned previously, F_2 and F_3 can be considered, respectively, as the input field and category field of another fusion ART neural network with a single input field only. Each node in F_3 represents an episode encoded as a pattern of sequential order according to the invariance principle in its weighted connections. The overall algorithm of episode learning can be described as Algorithm 2.

One important parameter used in the episode learning algorithm is ρ^s, the vigilance parameter in the F_2 field. The vigilance parameter is used as a threshold for the template matching process as described in Section IV-A.

B. Episode Retrieval

After the episodes are learned, any such episode can be recalled using various types of cues. A cue for the retrieval can be a partial sequence of any episode starting from the beginning or any position in the sequence. Based on the cue, the entire episode can be reproduced through the read out operation. An important characteristic of EM-ART is that the retrieval can be done in a robust manner as the activation and matching processes comprise analog patterns. This feature is useful when the cue for retrieval is imperfect or noisy. The approximate retrieval is also made possible by the use of fusion ART as the basic computational model for all parts of the EM. For example, lowering the vigilance parameter ρ^s of F_2 can make it more tolerant to noises or incomplete cues.

Algorithm 2 Episode Activation and Learning

1 FOR EACH subsequent event in episode $\mathcal{S}$
2 select a resonance node J in F_2 based on input $\mathbf{I}^k$ in F_1
3 let node activation $y_J \leftarrow 1$ (or a predefined maximum value)
4 FOR EACH previously selected node i in F_2
5 decay its activation by $y_i^{(\text{new})} = y_i^{(\text{old})}(1-\tau)$ or 0 if $y_i^{(\text{old})} \leq 0$
6 Given activation vector $\mathbf{y}$ formed in F_2 after the subsequent presentation of $\mathcal{S}$
7 select a resonance node J' in F_3 based on activation vector $\mathbf{y}$
8 learn its associated weight vector as $\mathbf{w}_J'^{(\text{new})} = \mathbf{y}$ if $\mathcal{S}$ is a novel episode

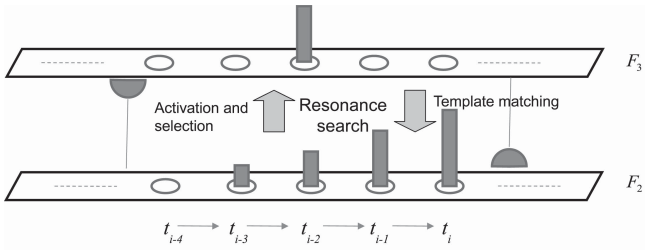

Fig. 5. Operations between F_2 and F_3.

To retrieve an episode based on a less specific cue, such as a subsequence of episode, a continuous search process is applied, in which the activity pattern of the cue is formed in F_2 while the F_3 nodes are activated and selected at the same time through the resonance search process. As long as a matching node is not found (still less than ρ^e), the next event is received activating another node in F_2 while all other nodes are decayed. For a cue as a partial episode, the missing event can mean no more new activation in F_2 while other nodes are still decayed. The algorithm for recognizing an episode based on imperfect cues can be described in Algorithm 3.

Once an episode is recognized, the complete pattern of sequence can be reproduced readily in the F_2 layer by the read out operation from the selected node in F_3 to the nodes in F_2. However, to reproduce the complete episode as a sequence of events, the corresponding values in F_1 layer must be reproduced one at a time following the sequential order of the events in the episode. EM-ART uses a vector complementing the values in F_2 before reading out the complete events in F_1. After the sequential pattern is read out to the field in F_2 which can be expressed as vector $\mathbf{y}$, a complementing vector $\overline{\mathbf{y}}$ can be produced so that for every element i in the vector, $\overline{y}_i = 1 - y_i$. Given the vector $\overline{\mathbf{y}}$, the node corresponding to the largest element in $\overline{\mathbf{y}}$ is selected first to be read out to the F_1 fields. Subsequently, the current selected element in the vector is suppressed by resetting it to zero, and the next largest is selected for reading out until everything is suppressed. In this way, the whole events of the retrieved episode can be reproduced in the right order.

Algorithm 3 Episode Recognition

1 FOR EACH incoming event
2 select a resonance node J in F_2 based on the corresponding event
3 let node activation $y_J \leftarrow 1$ (or maximum)
4 FOR EACH previously selected node i in F_2
5 decay its activation by $y_i^{(\text{new})} = y_i^{(\text{old})}(1-\tau)$ or 0 if $y_i^{(\text{old})} \leq 0$
7 select a resonance node J' in F_3 based on $\mathbf{y}$ in F_2
8 IF J' can be found THEN exit loop

C. Complexity Analysis

Consider the task of encoding m episodes with n unique events. We suppose each event contains a fixed set of t attributes and among the m episodes, there is a maximum of L events and an average of l events within each episode. In the following, we shall investigate the complexity of EM-ART in terms of the memory space and time required for learning and retrieving an episodes.

Space Complexity: As discussed in Section IV-B, for encoding n unique events, EM-ART requires n category nodes in the F_2 layer, each encoding an event as a multimodal pattern stored in the $2t$ weighted connections to the F_1 layer. Therefore, the total number of connections between the F_2 and F_1 layer is $2nt$. In addition, EM-ART encodes each episode as a F_3 category node fully connected to all the n F_2 nodes, where each weight to a F_2 node indicates the time position of the corresponding event within the episode. The exact encoding of these m episodes thus requires an EM-ART model with m nodes in F_3 layer and $2mn$ connections between F_2 and F_3 layers. Summing the space requirement for encoding events and episodes, the total number of nodes and weights in EM-ART model are $m+n$ and $2nt+2mn$, respectively. However, when generalization is allowed, the actual space requirement can be reduced by lowering the corresponding vigilance values (i.e., ρ^e and ρ^s).

Time Complexity: We identify comparison and multiplication to be the most critical operations during the encoding and retrieving processes in EM-ART. As described in Section IV-B, encoding an event requires a total of $O(nt)$ comparison operations during the resonance search process of fusion ART. After which, EM-ART averagely takes $O(lnt)$ processing steps to produce a series of activations in the F_2 layer, representing the order of the sequence to learn. Then, the activated F_2 pattern is matched against all patterns previously stored in the F_2-F_3 network and learned as a new category node in F_3 layer. Suppose there are already m episodes stored in this model, episode learning in the F_2-F_3 network takes a processing time of $O(mn^2)$. Therefore, the time complexity of encoding an episode in EM-ART is $O(ntl+mn^2)$. Similarly, consider an EM-ART with m episodes and n events, the time required to retrieve an episode is also in the order of $O(ntl+mn^2)$.

VI. Forgetting in Episodic Memory

Forgetting in episodic memory is essential to preserve and strengthen important and/or frequently used experiences, while removing unimportant or rarely occurred ones. Preventing ever-growing storage is a crucial aspect when dealing with continuous real-time operations. The forgetting mechanism should periodically check all stored events for their frequencies of use and the level of importance. Rarely-rehearsed events in episodic memory will be quickly forgotten while frequently-active ones will last longer.

In the proposed episodic memory model, a memory strength value $s_j \in [0,1]$, is associated with each event encoded by a F_2 node. Initially, s_j is set to s_{init} and gradually decays by decay factor $\delta_s \in [0,1]$. Upon an event reactivation, s_j is increased by an amount proportional to a reinforcement rate $r_s \in [0,1]$. The strength of an event e_j at time t can be computed as follows:

$$s_j(t) = \begin{cases} s_{\text{init}} & e \text{ is just created at } t \\ s_j(t-1) + (1-s_j(t-1))r_s & e \text{ is reactivated at } t \\ s_j(t-1)(1-\delta_s) & \text{otherwise.} \end{cases}$$

An event having s_j falling below a threshold $t_s \in [0,1]$ will be removed from episodic memory together with all of its weighted connections to/from other event and episode nodes.

The determination of parametric values on s_{init} and δ_s is mainly based on the nature of the associated application domain. Multiple values on these parameters can be included in one single episodic model. The various values on s_{init} and δ_s should be based on all related factors, such as rewards, prediction surprises, and emotions. Generally, the events with greater rewards, prediction surprises, and/or emotions should be stored in episodic memory for a longer time period. Hence, it should be associated with a higher value of s_{init} and/or a smaller value for δ_s.

VII. Benchmark Comparison

In this section, we compare the performance of the proposed model with other sequential memory methods to be discussed in Section IX for a word recognition task. In this task, we compare the performance of different models for the typoglycemia phenomena based on the following benchmark presented in [20]: "I cnduo't bvleiee taht I culod aulaclty uesdtannrd waht I was rdnaieg. Unisg the icndeblire pweor of the hmuan mnid, aocdcrnig to rseecrah at Cmabrigde Uinervtisy, it dseno't mttaer in waht oderr the lterets in a wrod are, the olny irpoamtnt tihng is taht the frsit and lsat ltteer be in the rhgit pclae. The rset can be a taotl mses and you can sitll raed it whoutit a pboerlm. Tihs is bucseae the huamn mnid deos not raed ervey ltteer by istlef, but the wrod as a wlohe. Aaznmig, huh? Yaeh and I awlyas tghhuot slelinpg was ipmorantt! See if yuor fdreins can raed tihs too."

To perform such benchmark test, each letter in the recognition test is fed into EM-ART model as an input vector one at a time. The input vector consists of 26 attributes, each of which represents a letter in the alphabet. At any time, only one attribute in the vector can be set to 1 to indicate the current letter read by the EM model. In the model, each letter is learned as an event node in F_1, while a unique word is encoded as an episode node in F_2 describing the ordering of its included letters (i.e., events).

We trained the EM model using all corresponding corrected words indicated by the typoglycemia test. With a vigilance of 1 at both the event and episode levels ($\rho^e = \rho^s = 1$), the model creates 26 event nodes and 73 episodes nodes, which corresponds to the 26 letters and 73 unique words in the typoglycemia test. After building the EM model, we load the test passage with all the misspelled words. Therefore, the model performance can be examined by the memory retrieval subject to the noisy cues with erroneous ordering.

We compare the performance with several other methods, including Markov chain [i.e., hidden Markov model (HMM)], Levenshtein distance method, and a spatio-temporal network model called LTM model, (i.e., long-term memory), as reported in [21]. LTM performs anticipation-based spatio-temporal learning that can store and retrieve complex sequences. Similar with EM-ART, LTM applies hierarchical network structure to encode complex events and their spatio-temporal sequences. To learn a text passage, a low-level neuron in LTM is used to learn a letter. The words can then be grouped and recognized into an upper-level neuron as a sequence of letters. As each typoglycemia word is forwarded as test inputs, the word learned by the neuron with the highest activation (i.e., highest similarity with the test word) is chosen as the prediction. The number of states and the minimum discrete density value is set to 6 and 10^{-4} as suggested by [21]. On the other hand, HMM method trains one HMM model for each unique word, wherein each observation symbol refers an unique letter in the alphabet. During typoglycemia test, the trained models are compared through their log-likelihood for each test word. The word with the highest likelihood is then retrieved as the predicted word. In this test, the HMM method is implemented based on the online toolbox available in [22]. The parameter setting follows those given in [23] for the word recognition tasks. Among the key parameters, the number of states and the minimum discrete density value are set to 6 and 10^{-4}, respectively. Another method compared in this test, the Levenshtein distance method, calculates the Levenshtein distance between each typoglycemia word with the learned words and retrieves the word with the minimum distance. Random selection is conducted if there are several learned words with the same minimum distance. The distance between two words in Levenshtein distance method is defined as minimum number of edits (replacement, deletion, or addition of a letter) needed to transform one word into the other. The space and time complexity of the various methods and models are shown in Table I. The deriving details are omitted due to the space limitation.

As shown in Fig. 6, although Levenshtein distance method requires the least space and processing time, it can only achieve an accuracy of 89.36% in retrieving the typoglycemia text. HMM can correctly retrieve 94.67% of the learned words from all words in the test. However, it requires intensive training of each word as one model. Both EM-ART and LTM models have 100% retrieval accuracy. By further comparing their space and time complexity, EM-ART employs less computational resources than LTM model to achieve the same level of retrieval performance as shown by Table I. These results show that EM-ART provides better recognition performance compared to HMM and Levenshtein distance method in the typoglycemia test. LTM has a similar performance as EM-ART by tolerating all errors while recalling the whole misspelled paragraph.

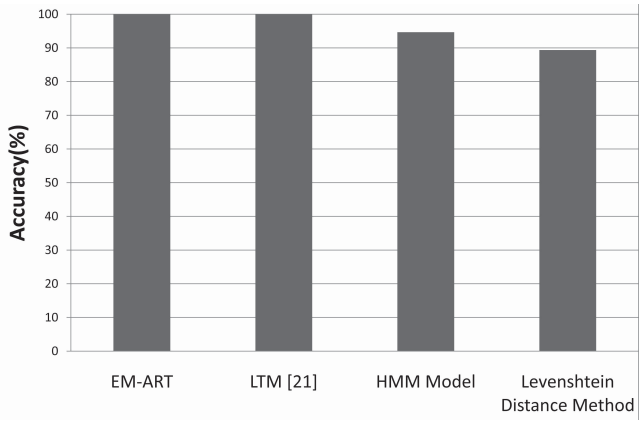

Fig. 6. Performance comparison on the word recognition benchmark.

TABLE I
COMPARISON OF SPACE AND TIME COMPLEXITY ON RECOGNITION BENCHMARK

	EM-ART	LTM	HMM	Levenshtein Distance
Space complexity	$O(nt+mn)$	$O(n^2t+mL^2)$	$O(mnl)$	$O(ml)$
Time complexity (learning)	$O(ntl+mn^2)$	$O(n^2tL+mL^3)$	mn^2l	NA
Time complexity (retrieving)	$O(ntl+mn^2)$	$O(n^2tL+mL^3)$	mn^2l	$O(ml^2)$

VIII. CASE STUDY IN A GAME DOMAIN

A. Episode Learning by a Game Agent

In this section, we study the performance of EM-ART in a first-person shooter game environment called Unreal Tournament (UT). In the UT environment, each nonplayer character (NPC) agent receives events describing the situation it experiences. The EM-ART model is used to learn episodic traces of those events, which are subsequently subjected to various recall tasks for performance evaluation. During the games, each NPC further embeds a reinforcement learning model based on [24] to guide its interactions with the game environment by continually optimizing the action policies, in a similar fashion to that of [25] and [26].

In our EM-ART model, an event can be represented as a vector, as shown in Fig. 2. Those events experienced by an agent during a battle, together with their mutual temporal relations, form an episode in the game. In this section, we investigate the experience of an agent from 100 battles (i.e., episodes) played in the game. During these 100 battles, there are 7735 events. The number of events within an episode varies from seven to over 250.

B. Episode Retrieval by a Game Agent

We build several exemplar EM-ART models using various vigilance values to access their effect on both episode learning and retrieval. Table II shows the memory sizes of the EM-ART models based on different vigilance setting, described by the total number of events and episodes in the built models. As reported in Table II, the size of memory shows almost no change as the vigilance at the episode level (i.e., ρ^s) drops from 1.0 to 0.9; meanwhile, a 0.05 decrease on event-level vigilance ((i.e., ρ^e) leads to a 60% reduction in the number of events by merging highly similar events into a single event node. The sensitivity of model over vigilance values reveals one remarkable characteristic of the UT domain—the similarity between events is relatively high, while most of exemplar episodes are distinct from each other.

After the EM-ART models are built, various tests are conducted to evaluate the accuracy of memory retrieval, subject to variations in cues, described as follows: 1) the cue is a full/partial event sequence of a recorded episode starting from the beginning/end/arbitary location of the episode and 2) the cue is a noisy or erroneous full length event sequence of the recorded episodes. In the retrieval test, the retrieval accuracy is measured using the ratio of the number of the correctly retrieved episodes over the total number of cues applied. We also further investigate the influence of different levels of vigilance on the model's performance at both the event and episode levels, indicated by the vigilance values of ρ^e and ρ^s, respectively. For the ease of the parameter setting, all our experiments use a standard vigilance value (ρ^e) throughout all the fields in the F_1 layer. We evaluated the performance of EM-ART under a range of vigilance values from 0.5 to 1.0 at both event (i.e., ρ^e) and episode (i.e., ρ^s) level. Due to the large amount and high similarity of results, in this paper, we only present the model performance under a narrower range of vigilance values from 0.9 to 1.0.

1) Retrieving from beginning of episodes: In this retrieval test, we extract partial sequences from the beginning of the recorded episodes as cues for retrieving the episodes. The cues are of different lengths, ranging from whole to 1/2, 1/3, 1/4, and 1/5 of the length of the episodes. Table III (a) shows the retrieval accuracy using cues of various length under different vigilance values. We observe that the model can accurately retrieve all stored episodes based on partial cues with different lengths with ρ^e of 1.0. With a lower ρ^e value of 0.95, the built model can still give a retrieval accuracy of 88% while reducing the number of encoded event nodes and connections by 60%. With $\rho^e = 0.95$, a higher retrieval accuracy can be typically provided by a longer cue. Meanwhile, the abstraction at the episode level (indicated by the value of ρ^s) shows an insignificant impact on the performance due to the data characteristics discussed previously.

2) Retrieving from end of episodes: In this retrieval test, cues are extracted from the tail of the recorded episodes. Similarly, cues of various length are used, ranging from whole to 1/2, 1/3, 1/4, and 1/5 of the original length of the episodes. Table III (b) shows the retrieval accuracy using cues of different length under various vigilance values of ρ^s. We see that the test shows similar performance patterns as those observed by retrieving from the beginning of episodes. Besides, the tests lead to an equal or better retrieval performance (i.e., at least 96% recognition rate) compared with retrieving from the beginning of episodes. The difference in performance can be observed by introducing the multiplicative decay process described in Section V-A. Given partial length cues from the beginning of episodes, this process tends to produce small differences between event activations and weighted connections encoded in the episode nodes.

3) Retrieving from arbitrary location of episodes: In this retrieval test, cues are extracted from the recorded episodes starting from randomly selected locations. Each such partial cue is forwarded to the model for episode retrieval. The cues are of different lengths, ranging from whole to 1/2, 1/3, 1/4, and 1/5 of the length of the episodes. Table III (c) shows the retrieval accuracy under various vigilance values. As indicated, the test provides similar retrieval performance as those by retrieving from beginning and end of episodes.

4) Retrieving with noisy events: To test the robustness of the model, we have further conducted the retrieval test with noisy data. Two types of errors are applied in the test as follows: 1) error in individual event's attributes and 2) error in event ordering within a complete sequence. This test investigates the model's robustness in dealing with the first type of noise. The corresponding noisy data set is directly derived from the original data set using the method described in Algorithm 4, with a specified error rate.

We test the model with various error rates on event representation and the results are shown in Table IV (a). The test shows that the built model can correctly retrieve at least 90% of all episodes with an error rate as high as 20%, at an event vigilance of 1. However, the performance drops to roughly 70% as the error rate reaches 30%. We further observe that, to achieve a high retrieval accuracy with noisy cues, the model requires a high vigilance ρ^e for event recognition in the F_2 layer, and the vigilance ρ^s for sequence recognition in the F_3 layer shows a relatively limited impact on the model performance. The results show that for event recognition, a higher vigilance

Algorithm 4 Generation of Noisy Events

Input: Error rate $r \in (0, 100)$
1 FOR EACH event in the original dataset
2 FOR EACH attribute a in the event
3 generate a random number, $rand =$ random number $(1, 100)$
4 IF $rand \leq r$, $a' = 1 - a$

TABLE II
EM MODEL SIZES AT VARIOUS VIGILANCES

(ρ^e, ρ^s)	Number of episodes	Number of events	Number of weights $F1 - F2(k)$	Number of weights $F2 - F3(k)$
(1.0, 1.0)	100	6705	509	1341
(1.0, 0.9)	100	6705	509	1341
(0.95, 1.0)	98	2692	294	527
(0.95, 0.9)	98	2692	294	527

TABLE III
ACCURACIES (IN %) OF RETRIEVING WITH INCOMPLETE CUES

Cue type	(ρ^e, ρ^s)	Cue length				
		Full length	1/2 length	1/3 length	1/4 length	1/5 length
(a) Partial cues from the beginning of episodes	(1.0, 1.0)	100	100	100	100	100
	(1.0, 0.9)	100	100	100	100	100
	(0.95, 1.0)	98	93	93	89	88
	(0.95, 0.9)	98	93	93	89	88
		Full length	1/2 length	1/3 length	1/4 length	1/5 length
(b) Partial cues from the end of episodes	(1.0, 1.0)	100	100	100	100	100
	(1.0, 0.9)	100	100	100	100	100
	(0.95, 1.0)	98	98	98	97	96
	(0.95, 0.9)	98	98	98	97	96
		Full length	1/2 length	1/3 length	1/4 length	1/5 length
(c) Partial cues from arbitrary location of episodes	(1.0, 1.0)	100	100	100	100	100
	(1.0, 0.9)	100	100	100	99	98
	(0.95, 1.0)	98	97	93	88	85
	(0.95, 0.9)	98	94	90	90	90

(ρ^e) is required to distinguish the highly similar but conceptually different events. In contrast, episode recognition should be able to tolerate minor changes within events and their temporal orders, which is achieved by lowering its vigilance (ρ^s). By setting appropriate vigilance values, the model tackles the challenge of building an efficient memory storage matching scheme as stated in Section II.

5) Retrieving with noisy episodes: In this section, we test the model reliability in dealing with the second type of noise. The corresponding noisy data set is derived from the original data set using the method described in Algorithm 5, given the desired rate of noise. In Algorithm 5, $S.e_i$ refers to the ith event within a stored episode/sequence S.

We test the model with various error rates on sequence representations and the results are shown in Table IV (b). To achieve tolerance to high level of noise, the model requires a relatively high event vigilance (ρ^e). With an event vigilance of 1, the model can achieve 100% retrieval accuracy with an error rate as high as 20%.

6) Retrieving with noisy events and partial episodes: We further investigate the retrieval performance of EM-ART by combining the two kinds of cue imperfections (i.e., partial length and noisy data) presented previously. To produce a set of noisy and partial cues, we generate two noisy data sets with 10% and 20% error on event representation as described in Algorithm 4. Then, the experiments from Section VIII-B1 to VIII-B3 are repeated with the vigilance level of $\rho^e = \rho^s = 1.0$. As shown in Table V, when more errors are added into the cues, EM-ART shows a degradation of retrieval performance across all types of partial cues. Generally, the performance difference using different types of partial cues widens as more noises are presented, in such a way that the performance is higher with longer cues and partial cues at the end of the episodes. This performance pattern is consistent with the results shown in Paragraph VIII-B-(a)–(c).

C. Comparison With a LTM Model

As EM-ART and the LTM (i.e., LTM) model [21] show a similar level of performance on the word recognition tests described in Section VII, we conduct further performance comparison between these two models by repeating the retrieval tests conducted in Section VIII-B. In addition, we investigate and compare the retrieval performance of the two models with an error rate of up to 50% in the cues at both the episode and event levels. The comparison between the two models is based on their retrieval accuracies with the best parameter settings tried. As the retrieval tests show that the performance of both models is almost the same with partial cues, only the retrieval results with noisy cues are presented in this paper.

TABLE IV
ACCURACIES (IN %) OF RETRIEVING WITH NOISY CUES

Cue type	(ρ^e, ρ^s)	Error Rate			
		5%	10%	20%	30%
(a) Full length cues	(1.0, 1.0)	97	97	92	76
with various level	(1.0, 0.9)	97	97	92	76
of noises on event	(0.95, 1.0)	96	95	87	63
representation	(0.95, 0.9)	96	95	87	63
		5%	10%	15%	20%
(b) Full length cues	(1.0, 1.0)	100	100	100	100
with various level	(1.0, 0.9)	100	100	100	100
of noises on sequence	(0.95, 1.0)	98	98	98	97
representation	(0.95, 0.9)	98	98	97	97

Fig. 7 shows the accuracy for retrieving episodes when noises are introduced to each individual event in an episode. Although both models provide lower performance as more noises are added, EM-ART can retrieve more correct episodes than the LTM model across all noise levels. While the LTM model shows a dramatic drop in performance as the error rate grows beyond 20%, EM-ART can still correctly retrieve at least 75% of the episodes with 30% noise. The performance difference between the two models, however, is most significant when dealing with noises in the event ordering. As shown in Fig. 8, a steady drop of accuracy level in the LTM model starts at 15% noise and continues with higher error rates. In contrast, the proposed model always successfully retrieves the correct episodes even though the error rate has reached 50%.

Algorithm 5 Generation of Noisy Episodes

Input: Error rate $r \in (0, 100)$
1 FOR EACH episode S in the original dataset
2 DO $rand$ = random number (1, $total\ number\ of\ episodes\ stored$)
3 WHILE $S_{\text{rand}} = S$ or $S_{\text{rand}}.length \leq \lfloor S.length * r/100 \rfloor$
4 x_1 = random number 1 to $(S.length - \lfloor S.length * r/100 \rfloor + 1)$
5 x_2 = random number 1 to $(S_{\text{rand}}.length - \lfloor S.length * r/100 \rfloor + 1)$
6 FOR $i = 0$ to $\lfloor S.length * r/100 \rfloor - 1$,
7 $S.e_{x_1+i} \leftarrow S(rand).e_{x_2+i}$

TABLE V
ACCURACIES (IN %) OF RETRIEVING WITH NOISY AND PARTIAL CUES

Cue type	Error rate	Full length	Cue 1/2 length	1/3 length	1/4 length	1/5 length
(a) Retrieval	10%	97	99	97	96	94
from beginning	20%	92	81	75	75	64
(b) Retrieval	10%	97	97	97	96	96
from end	20%	92	90	84	80	76
(c) Retrieval from	10%	97	96	97	93	90
arbitrary location	20%	92	91	82	79	64

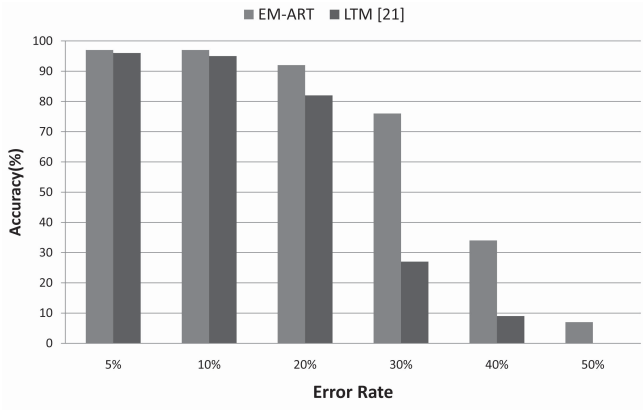

Fig. 7. Performance comparison for retrieving with various error rates on event representation.

The results above confirm that the proposed neural model for episodic memory can deal with imperfect cues and tolerate noises by doing approximate retrieval through the resonance search. The model is also more tolerant to noises and errors in memory cues than the LTM model.

D. Clustering Performance on Synthetic Sequence Data

In this section, we investigate the behavior of EM-ART in clustering sequence patterns using synthetic data sets. Each synthetic sequence data set consists of four groups of patterns, generated from four seed patterns using a specific method and distribution. The four seed patterns are denoted as "AJ," "JA," "KT," and "TK," where "AJ" for example consists of letters running to A to J in alphabetical order and "JA" consists of letters running from J to A in reverse alphabetical order. Among the four seed patterns, some patterns, for example "AJ" and "JA," have totally overlapping letters. However, certain patterns, for example "AJ" and "TK," do not share any letter at all. We generate a total of four synthetic data sets.

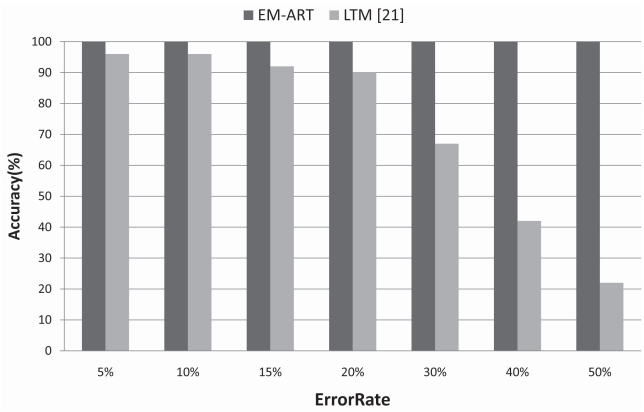

Fig. 8. Performance comparison for retrieving with various error rates on sequence representation.

The first two, called $e10$ and $e25$, are generated by toggling letters in the sequences with 10% and 25% probability, respectively, simulating errors in letter (event) recognition. The other two, called $s20$ and $s40$, are generated by switching the order of letter pairs in the sequences with 20% and 40% probability, respectively. For each data set, EM-ART is tasked to discover the groupings (clusters) under various vigilance settings.

The results as summarized in Table VI, show that, with a strict vigilance of 1, EM-ART creates one cluster for each distinct pattern. As the vigilance is relaxed, EM-ART starts to group similar patterns into clusters. At the extreme case, EM-ART groups all patterns into one cluster for vigilance equal or below 0.5. Somewhere in the middle, with a vigilance value of around 0.7, EM-ART is able to detect the original four clusters, tolerating the cluster distributions in $e10$ and $e25$. In comparison, EM-ART is significantly more robust in tolerating the variations in letter (event) sequences. With a fairly broad range of vigilance values from 0.8 to 0.95, EM-ART is able to detect the original four clusters, tolerating the cluster distributions in $s20$ and $s40$.

E. Analysis on Effects of Forgetting

Signals from environment are subject to noises. Through its one-shot learning, EM-ART encodes all the incoming events into its storage, regardless of the validity of the information. To deal with this problem, a forgetting mechanism is applied based on principles as follows: the noisy experiences are typically subjected to continuous decaying of their memory strength and eventually deleted from episodic memory due to low reactivation frequency, while a consistently happening experience tends to be preserved by high repetitions. In this way, not only does forgetting help episodic memory to maintain a manageable memory size in the long term, it also enhances the robustness and reliability of the model's performance in a noisy environment.

In this section, we simulate four sets of noisy training data as shown in Algorithm 6. In Algorithm 6, $\text{Set}_{\text{orig}}.e_i$ refers to the ith event within a dataset, Set_{orig}, based on the temporal sequence recorded. By setting r to 5, 10, 15, and 20, we generate four noisy data sets, each with 77 350 events and

Algorithm 6 Generation of Noisy Training Set

Input: Error rate $r \in (0, 100)$, original data set Set_{orig}, an empty data set Set_r and the total number of events (with duplication) n
Output: Data set Set_r
1 FOR $i = 1$ to n
2 Set_$r.e_i \leftarrow \text{Set}_{\text{orig}}.e_i$
3 FOR $j = 1$ to 4
4 FOR $k = 1$ to n
5 Set_$r.e_{j*n+k} \leftarrow \text{Set}_{\text{orig}}.e_k$
6 FOR EACH attribute a in the event Set_$r.e_{j*n+k}$
7 SET $rand =$ random number $1 - 100$
8 IF $rand \leq r$, $a' = 1 - a$

TABLE VI
NUMBER OF CLUSTERS AT DIFFERENT ρ^s

ρ^s	$e10$	$e25$	$s20$	$s40$	ρ^s	$e10$	$e25$	$s20$	$s40$
1.00	126	296	76	117	0.70	4	4	3	3
0.95	54	135	4	4	0.65	3	3	3	3
0.90	27	53	4	4	0.60	3	3	3	3
0.85	14	21	4	4	0.55	2	2	2	2
0.80	8	13	4	4	0.50	1	1	1	1
0.75	4	6	3	3					

1000 episodes. The generated data sets, respectively, containing 5%, 10%, 15%, and 20% errors on their event representation (and named, respectively, as Set_5, Set_10, Set_15, and Set_20) are used to train the episodic memory models. We then examine the performance of the trained models through retrieval tests, subject to various partial and noisy cues. Again, we measure the retrieval performance based on how many actual episodes can be correctly retrieved in a trial using the same type of cues. The performance is also compared with the original EM-ART without the forgetting mechanism.

We set the initial confidence $s_{\text{init}} = 0.5$, decay factor $\delta_s = 10^{-4}$, reinforcement rate $r_s = 0.5$, strength threshold $t_s = 0.1$, and vigilance $\rho = 0.5$ for event learning, and $s_{\text{init}} = 0.5$, decay factor $\delta_s = 0.008$, reinforcement rate $r_s = 0.5$, strength threshold $t_s = 0.1$, and vigilance $\rho = 0.95$ for episode learning. We train EM-ART for each generated training data set with a different level of noise. The memory size of the evaluated models is given in Table VII with comparison to their corresponding models without forgetting. From Table VII, we observe that as the error rate increases from 5% toward 20%, the evaluated models without forgetting have a larger number of event and episodes nodes by 66.7% and 9.8%, respectively. The significant increase on the memory size reflects the increased noises presented in the training sets. On the other hand, the models with forgetting show a marginal increase in their sizes due to continuous recognition of and thus deletion of noisy patterns.

After the models are built, we conduct various retrieval tests using noisy partial cues. Two exemplar sets of experimental results are presented with the following retrieval cues: 1) 1/3 noisy sequences of actual episodes starting from the end and

TABLE VII
COMPARISON OF MODEL SIZES AT VARIOUS ERROR RATES

Data set	Number of events	Number of events with forgetting	Number of episodes	Number of episodes with forgetting
Set_5	8635	7258	526	230
Set_10	10570	7809	570	231
Set_15	12505	8353	571	240
Set_20	14440	8907	578	241

2) 1/5 noisy sequences of actual episodes starting from the end. The noises on these cues are generated from the same set of actual experiences (i.e., D_{orig}) using Algorithm 4. The performance for these retrieval tests is compared with their counterparts without forgetting.

As shown in Figs. 9 and 10, forgetting helps episodic memory to retrieve more episodes correctly despite the reduction in memory size. The only exception is on the models built for Set_20, wherein retrieval with 1/5 length of the noisy sequences shows the same performance with or without forgetting. In general, longer cues provide a better performance for retrieval. As more noises are introduced, the model shows higher accuracies on retrieval both with or without forgetting. The difference in performance caused by forgetting also reduces as the error rate increases. This may be because higher noises tend to generate more distinct erroneous training samples and the original experiences can be retrieved more accurately.

IX. RELATED WORK

Many prior systems model episodic memory as traces of events and activities stored in a linear order, wherein some operations are designed specifically to retrieve and modify the memory to support specific tasks (e.g., [4], [27], and [28]). These approaches are limited to encoding simple sequential trace structure and may not be able to learn complex relations between events and retrieve episodes with imperfect or noisy cues. Although some models [4] have used statistical methods to deal with imperfect and noisy cues, they consider memory trace as a continuous series of events with no coherent representation of chunks of episodes as units of experience. Our proposed model addresses this issue by representing events as multichannel activation patterns allowing retrieval based on partial matching. Furthermore, the fusion ART fuzzy operations and the complement coding technique enable patterns to be generalized, so that irrelevant attributes of an event can also be suppressed through learning.

Another approach of episodic memory modeling uses the tree structure of a general cognitive architecture to store episodes instead of the linear trace (e.g., [5]). Each node in the memory tree includes some temporal information about its occurrence so that more complex representation can be expressed and episodes can be retrieved based on partial match. However, as it requires storage of every snapshot of the working memory, the system may not be efficient due to the possibly large storage of snapshots. In contrast, our episodic memory model clusters both individual events and

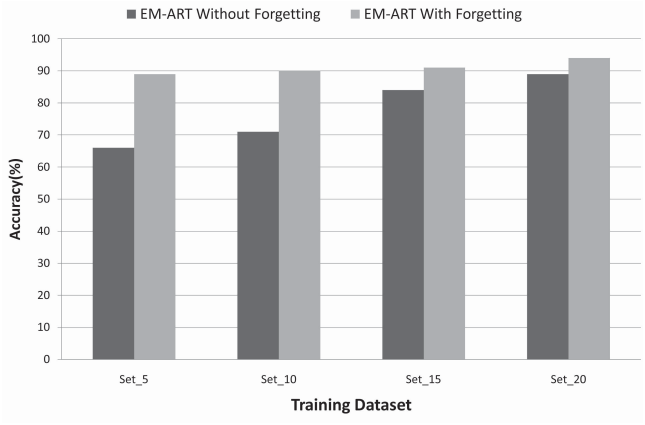

Fig. 9. Performance comparison for retrieving with 1/3 of episodes from end.

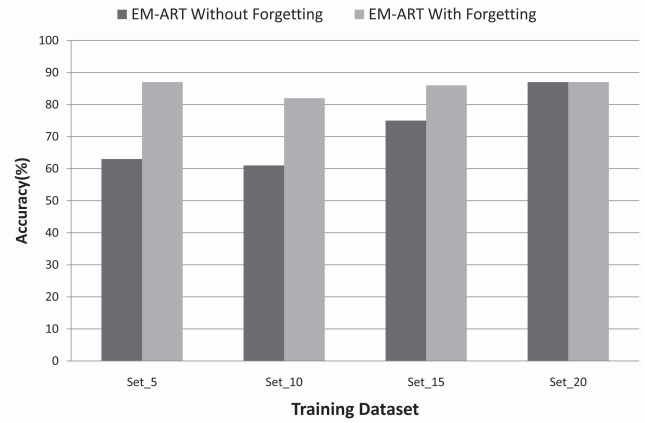

Fig. 10. Performance comparison for retrieving with 1/5 of episodes from end.

their sequential patterns based on similarities instead of holding all incoming information in a trace buffer. Our approach thus allows more compact storage and efficient processing.

On the other hand, most neural network models of episodic memory use associative networks that store relations between attributes of events and episodes (e.g., [6] and [7]). Although, they can handle partial and approximate matching of events and episodes with complex relationships, the associative model may still be limited in recalling information based on sequential cues. Some of the existing episodic memory models have attempted to address these challenges, in particular episode formation. Grossberg and Merrill combine ART neural network with spectral timing encoding to model timed learning in hippocampus [29]. Although, it can rapidly and stably learn timed conditioned responses in delayed reinforcement learning tasks, this model is only made specifically to handle learning timed responses but not other aspects of episodic memory, in particular, sequential ordering and multimodal association. SMRITI encodes events as relational structures comprised of role-entity bindings [8], without considering their spatio-temporal relations. Our proposed model tackles these issues by employing two levels of fusion ART. The first level deals with repetition by growing separate categories, while the second level clusters sequential patterns formed at the first level so that various lengths of complex sequential patterns can be learned at once. Our model thus is able to explore many possible complex relations, such as event and episode clustering as well as complex sequential learning. Another model called TESMECOR [30] captures complex spatio-temporal patterns and supports retrievals based on degraded cues. Using two neural layers consisting of nearly complete horizontal connections, the model distributively captures events and episodes without clustering. However, our approach offers modularity and flexibility by employing two levels of clustering that may be used by other systems.

X. Conclusion

We presented a new episodic memory model called EM-ART, based on a class of self-organizing neural networks known as fusion ART and the technique of invariance principle. Since EM-ART allows the memory to grow dynamically by allocating a new category node for each new pattern, EM-ART is able to encode and learn the episodes with variable length.

We conducted empirical experimental evaluation on EM-ART using a first-person shooting game, as well as a word recognition benchmark test. The experimental results showed that the model is able to provide a superior level of performance in encoding and recalling events and episodes even with various types of cue imperfections, including noisy and/or partial patterns. Our experiments on the synthetic data sets further revealed that EM-ART is especially robust in tolerating the variations in event sequences. In comparison, EM-ART does not generalize as well to noise at the event level. Finally, the experiments conducted also indicate that forgetting promotes an effective memory consolidation of its storage such that crucial knowledge can be kept in the memory, while the size of the stored information was regulated by discarding trivial and noisy information.

This paper has focused on the learning and retrieval functions within the episodic memory model. As discussed, episodic memory requires interactions with other related cognitive components to reveal its crucial roles. For example, the experiences stored in episodic memory may indicate more general knowledge in the form of a more permanent storage as semantic memory chunks. This indicates the potential of the co-evolving episodic-semantic model. Therefore, one immediate extension of our work is to explore its interaction with other memory systems, especially semantic memory.

References

[1] E. Tulving, *Elements of Episodic Memory*. New York: Oxford Univ. Press, 1983.
[2] M. A. Conway, "Exploring episodic memory," in *Handbook of Behavioral Neuroscience*, vol. 18. Amsterdam, The Netherlands: Elsevier, 2008, pp. 19–29.
[3] S. Zola-Morgan and L. R. Squire, "Neuroanatomy of memory," *Annu. Rev. Neurosci.*, vol. 16, pp. 547–563, Mar. 1993.
[4] S. T. Mueller and R. M. Shiffrin, "REM-II: A model of the developmental co-evolution of episodic memory and semantic knowledge," in *Proc. Int. Conf. Develop. Learn.*, vol. 5. 2006, pp. 1–7.
[5] A. Nuxoll and J. E. Laird, "Extending cognitive architecture with episodic memory," in *Proc. 22nd Nat. Conf. Artif. Intell.*, 2007, pp. 1560–1564.

[6] M. A. Gluck and C. E. Myers, "Hippocampal mediation of stimulus representation: A computational theory," *Hippocampus*, vol. 3, pp. 491–516, Oct. 1993.
[7] A. B. Samsonovvich and G. A. Ascoli, "A simple neural network model of the hippocampus suggesting its pathfinding role in episodic memory retrieval," *Learn. Memory*, vol. 12, pp. 193–208, Mar. 2005.
[8] L. Shastri, "Episodic memory and cortico-hippocampal interactions," *Trends Cognit. Sci.*, vol. 6, no. 4, pp. 162–168, Apr. 2002.
[9] A.-H. Tan, G. A. Carpenter, and S. Grossberg, "Intelligence through interaction: Toward a unified theory for learning," in *Proc. ISNN*, vol. 1. 2007, pp. 1094–1103.
[10] W. Wang, B. Subagdja, and A.-H. Tan, "A self-organizing approach to episodic memory modeling," in *Proc. Int. Joint Conf. Neural Netw.*, 2010, pp. 447–454.
[11] E. Dere, A. Easton, L. Nadel, and J. P. Huston, *Handbook of Episodic Memory*, vol. 18. Amsterdam, The Netherlands: Elsevier, 2008.
[12] L. Shastri, *Biological Grounding of Recruitment Learning and Vicinal Algorithms in Long-Term Potentiation* (Lecture Notes in Computer Science), vol. 2036. Berlin, Germany: Springer-Verlag, 2001, ch. 26, pp. 348–367.
[13] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern-recognition machine," *Comput. Vis., Graph., Image Process.*, vol. 37, pp. 54–115, Jan. 1987.
[14] A. Kaylani, M. Georgiopoulos, M. Mollaghasemi, G. Anagnostopoulos, C. Sentelle, and M. Zhong, "An adaptive multiobjective approach to evolving ART architectures," *IEEE Trans. Neural Netw.*, vol. 21, no. 4, pp. 529–550, Apr. 2010.
[15] K. S. Yap, C. P. Lim, and M. T. Au, "Improved GART neural network model for pattern classification and rule extraction with application to power systems," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2310–2323, Dec. 2011.
[16] D. Wang, B. Subagdja, A.-H. Tan, and G.-W. Ng, "Creating human-like autonomous players in real-time first person shooter computer games," in *Proc. 21st Annu. Conf. Innovat. Appl. Artif. Intell.*, 2009, pp. 1–6.
[17] S. Grossberg, "Behavioral contrast in short term memory: Serial binary memory models or parallel continuous memory models?" *J. Math. Psychol.*, vol. 17, pp. 199–219, Jun. 1978.
[18] G. Bradski, G. A. Carpenter, and S. Grossberg, "Store working memory networks for storage and recall of arbitrary temporal sequences," *Biol. Cybern.*, vol. 71, pp. 469–480, Oct. 1994.
[19] B. Subagdja and A.-H. Tan, "A self-organizing neural network architecture for intentional planning agents," in *Proc. 8th Int. Conf. Auton. Agents Multiagent Syst.*, vol. 2. 2009, pp. 1081–1088.
[20] G. E. Rawlinson, "The significance of letter position in word recognition," Ph.D. dissertation, Dept. Psychol., Nottingham Univ., Nottingham, U.K., 1976.
[21] J. Starzyk and H. He, "Spatio–temporal memories for machine learning: A long-term memory organization," *IEEE Trans. Neural Netw.*, vol. 20, no. 5, pp. 768–780, May 2009.
[22] *Hidden Markov Model (HMM) Toolbox for MATLAB*. (1998) [Online]. Available: http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html
[23] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
[24] A.-H. Tan, "Direct code access in self-organizing neural architectures for reinforcement learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2007, pp. 1071–1076.
[25] V. Vassiliades, A. Cleanthous, and C. Christodoulou, "Multiagent reinforcement learning: Spiking and nonspiking agents in the iterated prisoner's dilemma," *IEEE Trans. Neural Netw.*, vol. 22, no. 4, pp. 639–653, Apr. 2011.
[26] X. Xu, C. Liu, S. X. Yang, and D. Hu, "Hierarchical approximate policy iteration with binary-tree state space decomposition," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 1863–1877, Dec. 2011.
[27] W. C. Ho, K. Dautenhahn, and C. L. Nehaniv, "Comparing different control architectures for autobiographic agents in static virtual environments," in *Proc. IVA*, 2003, pp. 182–191.
[28] S. Vere and T. Bickmore, "A basic agent," *Comput. Intell.*, vol. 6, pp. 41–60, Feb. 1990.
[29] S. Grossberg and J. W. Merrill, "The hippocampus and cerebellum in adaptively timed learning, recognition, and movement," *Cognit. Neurosci.*, vol. 8, no. 3, p. 257–277, 1996.
[30] G. J. Rinkus, "A neural model of episodic and semantic spatiotemporal memory," in *Proc. 26th Annu. Conf. Cognit. Sci. Soc.*, Chicago, IL, 2004, pp. 1155–1160.

Wenwen Wang received the Bachelors degree in computer engineering from Nanyang Technological University, Singapore, in 2009, where she is currently pursuing the Ph.D. degree with the Center for Computational Intelligence.

Her current research interests include brain-inspired modeling of multimemory systems for agents.

Budhitama Subagdja received the Bachelors and Masters degrees in computer science from the Faculty of Computer Science, University of Indonesia, Depok, Indonesia, and the Ph.D. degree from the Department of Information Systems, University of Melbourne, Melbourne, Australia.

He is currently a Research Fellow with the School of Computer Engineering, Nanyang Technological University (NTU), Singapore. Before he joined NTU, he was a Research Assistant and a Lecturer with the University of Indonesia. He was a Post-Doctoral Fellow with the University of Melbourne, after finishing the Ph.D. degree. His current research interests include planning, reasoning, and learning mechanisms in autonomous agents and biologically inspired cognitive architectures for intelligent agents.

Ah-Hwee Tan (SM'04) received the Masters of Science and Bachelors of Science (Hons.) degrees in computer and information science from the National University of Singapore, Singapore, and the Ph.D. degree in cognitive and neural systems from Boston University, Boston, MA.

He is currently an Associate Professor and the Head of the Division of Software and Information Systems, School of Computer Engineering (SCE), Nanyang Technological University (NTU), Singapore. He has been a Faculty Member with SCE, since 2003, and was the Founding Director of the Emerging Research Laboratory, which is a research center for incubating new interdisciplinary research initiatives. Prior to joining NTU, he was a Research Manager with the A*STAR Institute for Infocomm Research, Singapore, spearheading the Text Mining and Intelligent Agents Research Programs. His current research interests include human-centric computing, brain-inspired intelligent agents, cognitive and neural systems, machine learning, knowledge discovery, and text mining.

Janusz A. Starzyk (SM'83) received the M.S. degree in applied mathematics and the Ph.D. degree in electrical engineering from the Warsaw University of Technology, Warsaw, Poland, and the Habilitation degree in electrical engineering from the Silesian University of Technology, Gliwice, Poland.

He was an Assistant Professor with the Institute of Electronics Fundamentals, Warsaw University of Technology. He spent two years as a Post-Doctoral Fellow and a Research Engineer with McMaster University, Hamilton, ON, Canada. Since 1991, he has been a Professor of electrical engineering and computer science with Ohio University, Athens, and the Director of Embodied Intelligence Laboratories. Since 2007, he has been the Head of the Information Systems Applications, University of Information Technology and Management, Rzeszów, Poland. He has cooperated with the National Institute of Standards and Technology, Gaithersburg, MD, in the area of testing and mixed signal fault diagnosis for eight years. His current research interests include embodied machine intelligence, motivated goal-driven learning, self-organizing associative spatiotemporal memories, active learning of sensory–motor interactions, machine consciousness, and applications of machine learning to autonomous robots and avatars.