3-2016

Semantic memory modeling and memory interaction in learning agents

Wenwen WANG

Ah-hwee TAN
Singapore Management University, ahtan@smu.edu.sg

Loo-Nin TEOW

Semantic Memory Modelling and Memory Interaction in Learning Agents

Wenwen Wang, Ah-Hwee Tan and Loo-Nin Teow

Abstract—**Semantic memory plays a critical role in reasoning and decision making. It enables an agent to abstract useful knowledge learned from its past experience. Based on an extension of fusion Adaptive Resonance Theory (fusion ART) network, this paper presents a novel self-organizing memory model to represent and learn various types of semantic knowledge in a unified manner. The proposed model, called FAMML, incorporates a set of neural processes, through which it may transfer knowledge and cooperate with other long-term memory systems, including episodic memory and procedural memory. Specifically, we present a generic learning process, under which various types of semantic knowledge can be consolidated and transferred from the specific experience encoded in episodic memory. We also identify and formalize two forms of memory interactions between semantic memory and procedural memory, through which more effective decision making can be achieved. We present experimental studies, wherein the proposed model is used to encode various types of semantic knowledge in different domains, including a first-person shooting game called Unreal Tournament, the Toads and Frogs puzzle, and a strategic game known as Starcraft Broodwar. Our experiments show that the proposed knowledge transfer process from episodic memory to semantic memory is able to extract useful knowledge to enhance the performance of decision making. In addition, cooperative interaction between semantic knowledge and procedural skills can lead to a significant improvement in both learning efficiency and performance of the learning agents.**

Index Terms—**semantic memory, learning agents, memory interactions, adaptive resonance theory**

I. Introduction

Semantic memory is a collective memory of concepts, facts, meanings, and other forms of general knowledge, that forms the basis of our understanding of ourselves and our environment independent from specific experience [27]. Given the facts that there is a wide variety of daily cognitive activities, e.g. reasoning, planning and remembering, that depend on the extensive store of semantic knowledge, semantic memory shows its centrality function to human's behaviors [2]. Through flexible retrieving, manipulating and associating the constituent facts and concepts from our semantic storage, we interpret our situated environment and interact accordingly in almost all of our daily activities. As an integral component of our long-term memory, semantic memory also plays its critical role in cognitive development with intensive interactions with other cognitive components, especially episodic memory and procedural memory. Specifically, the high level concepts and knowledge in semantic memory can be considered to be derived from the specific experiences stored in episodic memory [26]. In addition, semantic memory guides the reasoning and decision making functions in our daily life, and hence influences the development of the motor and cognitive skills in procedural memory [29].

In literature, many models have been proposed to study the underlying structure and mechanism to form and acquire semantic memory. The earlier works suggest to represent semantic knowledge in form of a set of abstract symbols [4], [11], [1]. These models provide dynamic encoding and manipulation of semantic knowledge by forming various types of propositions from their symbolic units. The learning of such semantic structures is accomplished by the statistical models, wherein the association/causality among concepts can be learned or inferred by their statistical co-occurences [15], [8]. However, these models are limited to certain pre-defined forms/types of semantic knowledge, such as the correlation among concepts. To establish a biological-plausible model of semantic memory, more recent development of connectionist studies represent semantic knowledge using a set of interconnected neural fields, wherein the information retrieval and inference is based on neural connections and activity propagation [6], [18]. However, the complexity of these networks raise a question on their piratical usage in real time applications. Given the fact that most of the existing semantic memory models represent and learn a selected form of semantic knowledge as a standalone memory component. Even those models, which focus on understanding the interactions between semantic memory and other memory components, are usually restricted to study some simple forms of interactions for specific tasks [15], [12], [20]. An additional central execution is usually required to oversee and regulate the memory interactions in a predefined and static manner.

In this paper, we present a novel memory model, namely FAMML (Fusion Adaptive Resonance Theory for Multi-Memory Learning) to encode and retrieve three main types of semantic knowledge using a unified set of computational principles and algorithms based on a class of self-organizing neural networks known as fusion Adaptive Resonance Theory (fusion ART) [24]. The model further incorporates a general learning procedure, wherein the contents of episodic memory may be consolidated and transferred to the more permanent form of semantic memory. In addition, we identify and formalize two major types of memory interaction between semantic memory and procedural memory, wherein semantic knowledge is used to support the acquisition of the procedural skills and improve the performance in decision making and problem solving.

Wenwen Wang and Ah-Hwee Tan are with the School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798; E-mail: {wa0003en, asahtan}@ntu.edu.sg.

Loo-Nin Teow is with DSO National Laboratories, 20 Science Park Dr, Singapore 118230; E-mail: tloonin@dso.org.sg.

In comparison with other existing work on semantic memory models, the main contributions and novelties of our work include (1) a generalized memory representation scheme for encoding a rich set of semantic memory based on multi-modal pattern association; (2) a unified set of learning operations for acquiring different types of semantic knowledge based on the dynamics of self-organizing neural networks; and (3) a framework for interaction across memory systems, specifically knowledge transfer from episodic to semantic memory and bidirectional interaction between the semantic and procedural memory, without an explicit executive control module.

We present experimental studies, wherein the proposed semantic memory model is embedded into autonomous agents to learn/encode various types of semantic knowledge in three applications. Firstly, the proposed model has been embedded into a learning agent in a first person shooting game called Unreal Tournament [30]. By consolidating the knowledge from the episodic memory to semantic memory, the experiments show that the learning agent is able to continuously acquire knowledge about the environment and improve its performance in reasoning and decision making. In the second domain, we illustrate how the proposed model can be applied to represent the various types of semantic knowledge in a puzzle game called Toads and Frogs, including general knowledge on move validity and game strategy. Compared with an agent using reinforcement learning only, our experiments show that the use of semantic knowledge leads to a higher success rate in solving the puzzle and a shorter time to reach the first success trial. In the final experiment domain, we incorporate semantic-procedural memory interaction into learning agents in a strategic game known as Starcraft Broodwar. The results show that the cooperative interaction between semantic knowledge and procedural skills can lead to a significant improvement in both learning efficiency and task performance.

The rest of this paper is organized as follows. Section II first provides a discussion and review of selected work on semantic memory models. Section III presents an overview for the multi-memory architecture, wherein the proposed semantic memory is represented, learnt and used through interactions with other memory systems. Section IV provides a brief summary of Fusion ART, which is used as a building block of the proposed semantic memory model presented in Section V. Section VI presents the memory consolidation process for abstracting semantic knowledge from an episodic memory model. Section VII provides the formalization and implementation details of the two types of memory interaction between semantic memory and procedural memory. Sections VIII presents the empirical evaluations of the proposed model in the three experimental domains. The final section concludes and highlights future work.

II. Related Work

Over the past decades, various types of structure and representation have been proposed to model semantic memory. In this section, we categorize and discuss the different types of the existing memory models based on the types of their individual knowledge representation as well as the learning and cognitive capabilities supported. In consideration of issues and challenges, we conclude the section with a motivation of our work.

A. Symbolic Models

One of the earliest models suggests that semantic memory stores simple logical propositions encoded as nodes and links of a *semantic network* [4]. The network explicitly expresses concepts and their interrelationships like “is a”, (e.g ”a flamingo is a bird”), “has”(e.g “a bird has wings”), or “can” (e.g “a flamingo can fly”) relations. A similar type of semantic memory network model called Fuzzy Maps (FCM) is also proposed to represent causality relationship between concepts by fuzzy logic [11]. Adaptive Control of Thought-Rational (ACT-R) cognitive architecture [1] represents a semantic memory as memory *chunk*s, wherein each chunk represents a concept by aggregating its properties and attributes of interest. It can evaluate the importance of a concept or a memory chunk through its general usefulness in the past. These symbolic models are rich in representation but do not focus on the learning of such semantic knowledge.

B. Statistical Models

Rather than simply representing meanings as symbolic relations, some models further incorporate statistical methods to support the discovery and learning of semantic knowledge. For example, Retrieving Effectively from Memory-II (REM-II) [15] and Topic [8] learn the relations between concepts by their individually defined statistical co-occurence indices. Some recent statistical models, e.g. predictive temporal context model (pTCM) [10] and Random Permutation Model (RPM) [19], further study the meaning of a word in a highly variable context by learning the association between the word and its time-drifting context represented by the group of adjacent words in the present sentence(s). Compared with symbolic models introduced previously, the statistical models support more robust knowledge retrieval, wherein partial or degraded retrieval cues can be handled by applying statistical inferences. Although these approaches allow learning, they only work for a very limited form of semantic memory, namely correlation between concepts.

C. Connectionist Models

Hinton [9] proposed one of the first neural network models on semantic memory, namely the Parallel Distributed Processing (PDP) model. It emulates the *semantic network* [4] by mapping its categorical representation into neural networks by representing concepts, relations, and properties with different neural fields. Beyond such representation, it also supports knowledge recollection and generalization through pattern completion accross the network. Rumelhart [21] further extended the PDP model to allow the automatical learning of relational and hierarchical structure for a semantic network. Using the backpropagation learning algorithm, the model can categorize and discriminate different concepts without direct supervision. However, Rumelhart’s model does not capture the temporal structure of the categorization process, which explains how the conceptual representations in a semantic structure can be dynamically shaped in real time. To tackle this challenge, semantic memory should incorporate a dynamic learning process to enable continuous formation of memory with the adaptive and gradual categorization and generalization.

Some more recent developments on fuzzy-neural models (T2-GenSoFNN and eFSM) [28] interpret semantic memory as a set of Mamdani-type IFTHEN fuzzy rules. By adopting an incremental learning and consolidation process of fuzzy rules, the fuzzy-neural models learn sets of evolving (time varying) semantic knowledge with the continuous and online streams of training samples fed by the environment. The models further employs a a novel parameter learning approach to tune the fuzzy set parameters in a real-time manner. However, as the complexity of the learning semantic knowledge increases (in terms of the number of network input attributes), the models may suffer from slower convergence time while it continuously learns in a dynamic environment.

On the other hand, some connectionist models focus on the biological details of the underlining neural structure which forms the basis of semantic memory in brain. In general, their studies indicate one key feature of the semantic memory. That is, semantic memory is not a monolithic unitary model but may involve multiple representation and learning mechanisms. Farah and McClelland [6] suggested a bidirectional network model consisting of different interconnected neural fields to represent the corresponding sensory-functional features of semantic concept or knowledge. The model is developed further as the convergence theory of semantic memory in which more perceptual and functional features like actions, sounds, and olfactions are incorporated as different neural fields [18]. However, due to the computational complexity, these models can hardly be implemented and therefore have limited usage in piratical domains.

D. Models on Semantic and Episodic Memory Interactions

Most semantic models mentioned above are studied as isolated memory systems that process and acquire semantic knowledge directly from the inputs. However, some models also consider episodic memory to be attached to semantic memory to form an integrated memory system. For example, REM-II [15] connects episodic memory and semantic memory to learn statistical relationships between items within and across time. Another episodic memory model based on the SOAR cognitive architecture [12] embeds episodic memory directly into the symbolic semantic memory model as additional properties providing contextual and historical information of each assertion and update in the memory. A distributed approach called TESMECOR [20] also considers episodic memory as distributed neural connections that also support semantic representation. Although these integrated approaches may provide workable mechanisms to store and retrieve knowledge based on both temporal and relational structures, they do not aim to account for neuropsychological data of semantic memory.

In contrast, some biologically inspired works study dual memory systems by aligning their proposed model with the relevant neuropsychological findings. The cortico-hippocampal neural model [7] suggests that hippocampus (episodic memory) and neocortex (semantic memory) are two parallel memory systems receiving the same input. The hippocampus learns an internal representation to associate the input and recalled patterns, while the neocortex categorizes the input based on the internal representation formed by the hippocampus. In this way, episodic memory and semantic memory can work together for both semantic categorization and episodic information retrieval. A more realistic model of episodic-semantic memory interaction called Complementary Learning Systems (CLS) [17] mirrors the network structure and connections between hippocampus and neocortex in the brain and incorporates a memory consolidation process. Based on neuroscientific evidences that neurons in hippocampus are reactivated spontaneously during slow wave sleep [31] and thus reinstating the patterns in neocortex to enact slow incremental learning, CLS also emulates an offline consolidation process by randomly reactivating memory recollection in hippocampus to be used as inputs for neocortex. A more recent work on Competitive Trace Theory (CTT) [33] further extends the ideas from CLS model with a novel decontextualization process during the memory consolidation from episodic events to semantic traces. In the proposed decontextualization process, the memory trace is reactivated repeatedly along time. While the core/overlapping features on the similar events are strengthen to form their common and semantic representation, the non-overlapping/context features among these episodic events mutually inhibit each other such that none of them can be retrieved and hence "decontextualized" from their semantic representation.

Although the complementary neocortex and hippocampus models above provide a general framework that can be confirmed by the evidences of memory consolidations and lesions behavior in the brain, they do not provide a computational account of the key processes of learning and memory interaction, specifically the consolidation process from episodic memory to semantic memory and the role of semantic memory in decision making and learning. .

E. Models on Semantic and Procedural Memory Interactions

Some semantic models have combined with a model of procedural memory in parallel to investigate their different roles in learning and decision making. In general, most of the existing works (e.g.[16], [12], [22]) on semantic-procedural dual memory systems focus on a single form of interaction, wherein the semantic memory provides the necessary reasoning to activate the relevant procedural knowledge. This simplified interaction process limits the usage of semantic memory by activating semantic memory only in cases with insufficient procedural knowledge. More realistic models of interactions should be proposed to allow better utilization on the entire knowledge base consisting of both semantic and episodic memory. In this work, we should conduct an in-depth study into how the interaction enables the model to produce a more versatile capability in decision making and problem solving.

Sun and Mathews proposed the use of an executive control [23] to monitor and control the interactions between the two long-term memory systems. The executive control module used a set of predefined rules to explicitly regulate the interaction process. However, this raises the question of how the executive control knowledge can be acquired through a learning process and how it can be continuously refined over time.

F. Summary

We summarize our review of the existing work on semantic memory modelling in this section. As shown in Table I, typical statistical models allow learning but only work for a limited form of semantic memory, namely correlation between words and concepts. On the other hand, connectionst models, which explore to encode and manipulate a wider range of semantic knowledge with the multiple-modal/field representation, suffer from potentially long convergence time and hence have a limited usage in real time domains. In literature, the studies of memory interactions have been limited to a single form of interactions within a dual memory systems (either semantic-episodic or semantic-procedural interactions), wherein a central executive module is also employed to regulate and guide the interactions. In view of the dilemma between learning and richness in semantic representation, our work aims to derive a coherent set of principles and processes to represent and learn a sufficiently rich set of semantic knowledge. Specifically, the proposed principles and processes should be general enough to allow the encoding and learning of various types of semantic memory in a single unified manner. Considering the key issues in modelling semantic memory, the proposed model is designed to tackle the challenges in the following: (1) effective knowledge representation for encoding various types of concepts and their possible relations, including concept hierarchy, association rules and casual relations; (2) online and incremental semantic learning process, wherein semantic knowledge can be learned and evolved in response to a continuous streams of inputs from environment; (3) collaborative interaction processes with both semantic, episodic and procedural memory to facilitate decision making and reasoning. The interaction should preferably be based on a self-organizing mechanism using the neural connections and activity propagation, without direct control from a central executive control system.

III. The Multiple-Memory Architecture

To model the semantic memory and its relations with other long-term memory systems, we first present a neural network-based cognitive model known as FAMML with an explicit modelling of semantic memory, procedural and episodic memory systems. The design of FAMML is consistent with the structure of the cortico-hippocampal neural model proposed by McClelland et al. [13]. More importantly, a full computational realization of the multi-memory systems is provided based on self-organizing neural networks.

For ease of discussion, FAMML consists of the minimal structure sufficient to illustrate the roles of semantic memory in a learning agent. As shown in Figure 1, FAMML contains five main components, described as follows.

- **Intentional Module** maintains a set of goals in hand to regulate the decision making process linking sensory input to motor responses.
- **Working Memory** is a limited-capacity memory buffer that maintains all the necessary information and knowledge online for use in performing the current task.
- **Semantic Memory** encodes various forms of semantic knowledge, including concept hierarchy, causal relations and association rules.
- **Episodic Memory** stores the specific past experiences in the form of events and episodes of the spatio-temporal relations among events.
- **Procedural Memory** contains a collection of action rules, which encode the sequences of situation-action pairings to perform the familiar routines and other well-rehearsed tasks.

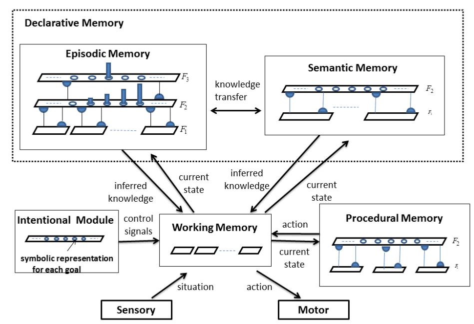


Fig. 1: An illustration of FAMML model.

In this architecture, each of long-term memory modules is capable of performing its own operations, including encoding, learning, and retrieval of memory, independently from the other components. However, the decision process and functions of the overall system is a result of the complex interactions among the various memory modules. The main memory operations of semantic memory can be listed as follows:

- **Memory consolidation through playback of episodic memory**. At some point in time, the contents of episodic memory are read out to the working memory which starts the learning process in semantic memory.
- **Semantic memory retrieval by pattern completion**. The semantic knowledge can be retrieved by providing memory cues as a subset or portions of the target information to be retrieved.
- **Action selection and behavior learning by procedural memory**. At any point in time, the current action to take is selected through the procedural memory by executing the fired procedural rule. During the process of online decision making, the agent can discover the solutions to novel situations and tasks based on the knowledge and reasoning provided by semantic memory. The new situation-action pairings can be further written to procedural memory based on its frequently successful attempts.

TABLE I: Comparison between semantic memory models

Model	Approach	Representation	Learning Method	Memory Interaction
REMII [15]	statistical	concurrence matrix of features	statistical learning on the co-occurences between features	episodic to semantic
pTCM [10]	statistical	vectors of features	associates each semantic representation with a inner drifting context	none
eFSM [28]	connectionist	Mamdani-type IFTHEN fuzzy rules	incremental learning and pruning on fuzzy rules and fuzzy parameters	none
CLS [17]	connectionist	two-layer network with Conditional Principal Components Analysis (CPCA) Hebbian learning	distributed neural patterns of semantic concept	episodic to semantic
CLARION [22]	hybrid	two level networks with declarative chunks and association rules in top level and auto-associative memory in bottom level	top: general knowledge store (GKS); bottom: backprorogation learning	bi-directional interaction between procedural and semantic memory under the supervision from executive control

IV. Fusion ART

Our proposed semantic memory model is built based on the multi-channel self-organizing fusion ART neural networks. Extended from Adaptive Resonance Theory (ART) [3], fusion ART offers a set of universal computational processes for encoding, recognition, and reproduction of patterns. Given an input pattern, fusion ART conducts a bi-directional search for best matching pattern. It then updates the connection weights and learns the patterns upon each successful matching. In the case of mismatch, a new category node is dynamically added to learn the new pattern presented. By applying fuzzy operations and *complement coding* [24], fusion ART dynamically performs pattern generalization. This type of neural network is chosen as the building block of our memory model as it enables continuous formation of memory with an adjustable vigilance of categorization to control the growth of the network and the level of generalization.

Figure 2 illustrates the fusion ART architecture, which may be viewed as an ART network with multiple input fields.

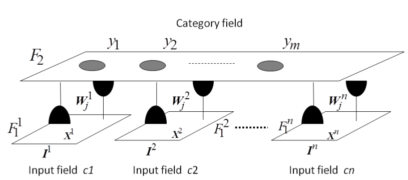


Fig. 2: The fusion ART architecture.

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

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 where $i > 1$ indicate that it is the ith field. The standard multi-channel 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. Initially, F_2 contains only one *uncommitted* node and its weight vectors contain all 1's.
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 multi-channel 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

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

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

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

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. When an uncommitted node is selected for learning a novel pattern, it becomes *committed* and a new uncommitted node is added to the F_2 field. Fusion ART thus expands its network architecture dynamically in response to the input patterns.

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

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

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 $\mathbf{x}^{k(\text{new})} = \mathbf{w}_J^k$.

V. Semantic Memory Representation

In this research, we view that the semantic memory is not unitary. In other words, there may be different types of semantic memory networks, each representing a different knowledge structure. A mathematical formulation for the representation of semantic memory is presented as follows.

Semantic Memory, denoted by $S = \{S_1, S_2, ...\}$, can be viewed as a set of semantic fragment or rules. Each semantic rule S_i can be one of the three basic types described as follows: (1) A rule of concept hierarchy defines the "IS-A" relation between two known concepts and can be represented by $S_i = s_a : s_A$, wherein s_a and s_A refer to the memory representation of the concept a and its category A respectively (e.g."Pigeon is a kind of bird."); (2) an association rule indicates the co-occurrence of two memory states, each representing any piece of information or concept stored. Each association rule is represented as $S_i = (s, s')$, where s and s' indicate the two associated objects or concepts (e.g. "People who buy milk usually buy some bread together."); (3) a causal relation rule states the causality between two memory states and is written as $S_i : s \rightarrow s'$, wherein s refers to the cause and s' represents the effect (e.g. "Eating crabs with some fruits usually causes diarrhoea and vomiting.").

For semantic knowledge representation, each input field of the fusion ART represents a concept or a relevant piece of information for a semantic rule of interest. Taking in the corresponding input vector $\mathbf{I}^k$ for the input field F_1^k, each part (s_a,s_A, s or s') of the semantic rule S_i can be learned as the weight vector $\mathbf{w}_j^k$ through the fusion ART operations as described in Section IV. Hence, each semantic rule S_i is encoded as a category node in F_2 by encoding the proper relations ("IS-A" , associative or causal) among all the underlying concepts or information fragments as s_a,s_A, s or s'. Figure 3 illustrates how fusion ART is used to represent various types of semantic memory. Each type of semantic memory can be made as a single fusion ART network.

From our formulation of the semantic memory, each entry in semantic memory generalizes similar inputs into the same category rather than as separate entries. The generalization can be achieved by lowering the vigilance parameter ρ so that slightly different input patterns will still activate the same

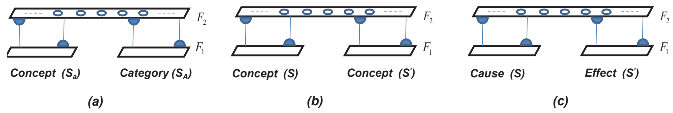


Fig. 3: The three types of semantic knowledge and their neural network representations: *(a)* concept hierarchy $\mathbf{s_a} : \mathbf{s_A}$; *(b)* association rule $(\mathbf{s}, \mathbf{s}')$; *(c)* causal relation $\mathbf{s} \rightarrow \mathbf{s}'$.

category. The learning process for various types of semantic memory is given in Section VI.

VI. Knowledge Transfer from Episodic Memory

Prior to the discussion on the learning process of semantic knowledge via episodic memory, a mathematical formulation for episodic memory is introduced as follows.

Episodic Memory, denoted by $E = \{E_1, E_2, ...\}$, stores the specific experiences in the past. Each episode E_i is an ordered sequence of events such that $E_i = [e_{t_1}, e_{t_2}, ..., e_{t_m}]$, where t_j denotes the relative time point wherein the event e_{t_j} occurs. The event sequence for each episode E_i is timely ordered such that $t_1 < t_2 < ... < t_n$. An event e_{t_j} describes the snapshot of experience at time t_j such that $e_{t_j} = (\mathbf{v}_{t_j}^{c1}, \mathbf{v}_{t_j}^{c2}, ..., \mathbf{v}_{t_j}^{ck})$ where each vector $\mathbf{v}_{t_j}^{ck}$ contains a set of attributes to describe a critical aspect c_k of the event. Some examples of frequently used event fields are *location*, *time*, one's own *internal state*, *observed external states*, *action*, *consequence/reward*.

As discussed in Section I, semantic memory can be learnt through a gradual knowledge transfer process from episodic memory, wherein the contextual information and the association with the specific experience are removed to form the corresponding general (semantic) knowledge [27].

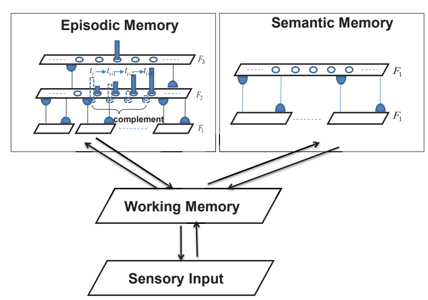


Fig. 4: Pathways for knowledge transfer from episodic to semantic memory.

Figure 4 illustrates the interaction pathways supporting the transfer of information from episodic memory to semantic memory. During knowledge transfer, the episodic memory recalls and readouts the stored episodes in a memory playback process [30]. During the recall of each episode, each associated event in the episode is reproduced as the output of the episodic memory (via F_1) and activates the corresponding regions in the working memory. For learning causal rules, the ordering of the event reproduction should be consistent with that of the temporal information stored (via weights of F_2). As each event is presented, it is reevaluated for whether it describes an instance for a semantic rule of the interested type. If the presented event describes an experience of interests, the current state s of the working memory is presented to semantic memory to form a training sample for learning the specific semantic knowledge. Otherwise, the memory playback continues with the next stored event. The detailed process of knowledge transfer from episodic to semantic memory is summarized into Algorithm 1.

Algorithm 1 Process for knowledge transfer from episodic memory (EM Transfer)

for each episode E stored in the episodic memory network **do**
    for each event e within episodic E **do**
        update the current state s of the working memory based on the event pattern of e
        for each semantic network S currently available **do**
            if the current state s describes an instance of semantic knowledge learned by S **then**
                update the input field F_1 of S
                activate and select a node (through winner-take-all) in F_2 of S
                while the node is not in resonance or the node has been selected previously **do**
                    reset the current node activation
                    choose another node in F_2 of S
                end while
                if no matching node can be found in F_2 of S **then**
                    recruit an uncommitted node in F_2 of S
                    learn it as a novel piece of semantic knowledge
                end if
            end if
        end for
    end for
end for

VII. Semantic and Procedural Memory Interaction

In this paper, we focus on two basic types of interaction between the semantic and procedural memory. Prior to discussing the processes of interactions, we first present a mathematical formulation for procedural memory, as follows.

Procedural Memory, denoted by $P = \{P_1, P_2, ...\}$, is a set of action rules which perform the familiar tasks and routines. Each action rule P_k suggests a possible action a with a certain level of expected reward r (payoff), based on a given situation s. Therefore, each action rule can be represented as $P_k : s \rightarrow (a, r)$. Through reinforcement learning, procedural memory learns the association of the current state and the chosen action to the estimated reward. In this work, our procedure memory network is implemented based on a three-channel fusion ART model, also known as the TD-FALCON network presented in our prior work [25].

In order to learn the procedural knowledge, the procedural memory model comprises a cognitive field F_2^c and three input fields, namely a sensory field F_1^s for representing current states, an action field F_1^a for representing actions, and a reward field F_1^r for representing reinforcement values. Designed as a fast and incremental learning method, TD-FALCON has been shown to outperform standard Q-Learning as well as many gradient-descent based reinforcement learning algorithm [25], [32]. In view of space constraint, a summary of the TD-FALCON reinforcement learning is presented as Algorithm 2 below.

Working together, the semantic memory and procedural memory constitute the knowledge base of the cognitive architecture, which guides the behaviors of the agent through their interactions. Figure 5 provides an overview of their interaction pathways. In the rest of the section, we present two main types of semantic and procedural memory interactions.

Algorithm 2 TD-FALCON Reinforcement Learning in Procedural Memory

initialize the Procedural Memory network
repeat
    sense the environment and update current state s in Working Memory
    for each available action a **do**
        predict the reward r by presenting s and a to the network
    end for
    based on computed reward values, select action a with highest reward
    perform action a, observe next state s' and receive a reward r from environment
    compare estimated reward $Q(s, a)$ using a temporal difference method
    present the corresponding state, action and the updated reward estimation, namely s, a and $Q(s, a)$, to learn the procedural rule as $P_{new} : s \rightarrow (a$,Q(s,a))
    update the current state by $s = s'$
until goal is realized or s is a terminal state

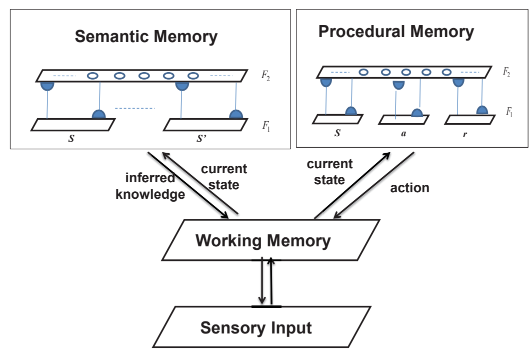


Fig. 5: Interaction pathways for semantic and procedural memory interactions.

A. Semantic to Procedural Interaction

In this form of interaction, semantic memory is used to provide the contextual information in order to activate the relevant action rules in the procedural memory. More formally, the interaction involves the flow of information from semantic memory to procedural memory, as defined below.

Definition 1 (SP Interaction): Given the current state s in working memory, a threshold τ, and the following knowledge fragments from semantic memory and procedural memory:

$$\begin{cases} S_i : s \rightarrow s' \quad or \quad S_i = (s, s') \quad or \quad S_i = s : s' \\ P_k : s' \rightarrow (a, r) \\ r \geq \tau \end{cases}$$

where semantic memory indicates that the current state s usually associate with (or lead to) another state s'. s' can trigger an action a leading to a good outcome according to the procedural rule P_k

Upon SP interaction, if the procedural rule indeed leads to a favorable outcome, the procedural memory may learn to directly associate the memory state of s with action a, which can be expressed as: $P_{new} : s \rightarrow (a, r)$.

Although the semantic and procedural memory modules should run in parallel, the complete process of SP interaction and transfer can be implemented with a sequential algorithm as presented below.

Algorithm 3 Reinforcement Learning with Semantic-Procedural Interaction

initialize the Procedural Memory
repeat
  sense the environment and update current state s in Working Memory
  repeat
    set input vector $\mathbf{I} = s$ as memory cue of Semantic Memory
    Semantic Memory retrieves the most relevant semantic chunk in the form of
    $S_i : s \to s'$, $S_i = (s, s')$ or $S_i = s : s'$
    update Working Memory as s'
    update its sensory channel F_1^s of Procedural Memory by setting $I^s = s'$
    Procedural Memory searches for a procedural rule $P_k : s' \to (a, r)$ based on the updated F_1^s field
  until a procedural rule is fired or time-out
  if a procedural rule is fired **then**
    perform the identified action a
  else
    perform a random action
  end if
  compare estimated reward $Q(s, a)$ using a temporal difference method
  present the corresponding state, action and the updated reward estimation, namely s, a and $Q(s, a)$,
  to learn the procedural rule as $P_{new} : s \to (a,\text{Q(s,a)})$
  update the current state by $s = s'$
until goal is realized or s is a terminal state

B. Procedural to Semantic Interaction

For making a decision, procedural memory may explicitly prime the semantic memory for the unknown information and knowledge for firing a specific action rule. The search in the semantic memory can be triggered by rising the attention levels for those missing attributes in the working memory. More formally, the interaction involving the flow of directive signals from procedural to semantic memory is defined as follows.

Definition 2 (PS Interaction): Given the current state s and a procedural rule

$$P_k : s' \to (a, r),$$

the semantic memory is primed to search for semantic knowledge of the form:

$$S_i : s \to s' \quad or \quad S_i = (s, s') \quad or \quad S_i = s : s'$$

which will lead the current state from s to s'. If S_i is found, the procedural rule P_k is fired.

Upon PS interaction, if the selected procedural rule leads to a favorable outcome, the procedural memory may learn to directly associate the memory state of s with the action a as $P_{new} : s \to (a, r)$. This interaction and transfer process can be embedded into a reinforcement learning algorithm presented in Algorithm 4.

VIII. Empirical Evaluation

A. Non-player Character Modelling in Unreal Tournament

For performance evaluation, FAMML is embedded into an autonomous non-player character (NPC) agent playing the Unreal Tournament (UT) game. The experiments using UT are conducted to see if FAMML can produce useful knowledge

Algorithm 4 Reinforcement Learning with Procedural-Semantic Interaction

initialize the Procedural Memory network
repeat
  sense the environment and update current state s in Working Memory
  repeat
    Procedural Memory searches for a procedural rule $P_k : s' \to (a, r)$
    Semantic Memory searches for a semantic rule $S_i : s \to s'$, $S_i = (s, s')$ or $S_i = s : s'$ by setting the input vector as $\mathbf{I} = (s, s')$
    updates Working Memory as s'
  until a procedural rule is fired or time-out
  if a procedural rule is fired **then**
    perform the identified action a
  else
    perform a random action
  end if
  compare estimated reward $Q(s, a)$ using a temporal difference method
  present the corresponding state, action and the updated reward estimation, namely s, a and $Q(s, a)$, to learn the procedural rule as $P_{new} : s \to (a,\text{Q(s,a)})$
  update the current state by $s = s'$
until goal is realized or s is a terminal state

and improve the performance of the agent. The scenario of the game used in the experiment is "Death match", wherein the objective of each agent is to kill as many opponents as possible and to avoid being killed by others. In the game, two (or more) NPCs are running around and shooting each other. They can collect objects in the environment, like health or medical kit to increase its strength and different types of weapon and ammunition for shooting.

In our experiments, all agents (that we evaluate) play against a baseline NPC agent called *AdvanceBot* that performs according to a set of hard-coded rules. There are four different hard-coded behavior modes in *AdvanceBot* (i.e. running around, collecting items, escaping away and engaging in battle). *AdvanceBot* always chooses one of the four behaviors based on a set of predefined rules. Under the battle engagement behavior, the agent always selects the best weapon available for shooting based on some heuristics optimized for a certain environment map used in the game.

1) Memory Enhanced Agents: To investigate how the proposed semantic memory module contributes to the overall agent performance, different agents with different memory configuration embedded are tested and compared. This experiment employs two memory-based agents, namely *RLBot* with procedural memory embedded and *MemBot* incorporating the full integrated procedural-declarative memory system.

a) Agents with Procedural Memory: The agent embedding procedural memory module (i.e. *RLBot*) learns and performs its behavior selection through the TD-FALCON reinforcement learning algorithm as stated in Algorithm 2. The state, action, and reward vectors in Figure 6 correspond to the input representation in the three ART network of *RLBot*. Specifically, behavior pattern (i.e. running around, collecting items, escaping away and engaging in battle) in the state vector represents the behavior currently selected. The action vector indicates the next behavior to be selected. Based on the state vector and the reward vector (set to the maximum value), the network searches the best match category node and reads out the output to the action field indicating the behavior type to

TABLE II: Sample procedural rules learnt in UT

```
IF health is around 19, and not being damaged,
   and not seen enemy,
   and has adequate ammo,
   and currently in running around state;
  THEN go into collecting items state;
  WITH reward of 0.556.
```

be selected. The network then receives feedbacks in terms of the new state and any reward given by the environment.

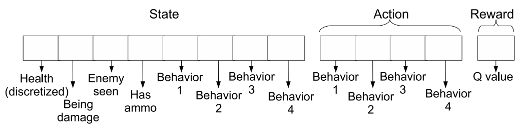


Fig. 6: State, action and reward representation for procedural memory model.

The network learns by updating the weighted connections according to the feedback received and applying temporal difference methods as described in [25] to update the reward field. The agent receives the reward signal (positive or negative) whenever it kills or is killed by another agent. In this way, *RLBot* continually learns and acquires procedural knowledge on behavior selections (as illustrated by the sample rule shown in Table II) while playing the games. In contrast to *AdvanceBot*, *RLBot* chooses an available weapon randomly in the battle engagement behavior. Another agent called *RLBot++* is also used to employ the same reinforcement learning model as *RLBot* but select the weapon based on the optimized predefined rules just like in *AdvanceBot*.

b) Agent with Procedural-Declarative Memories: The proposed declarative model is embedded into an agent (i.e. *MemBot*), which has the same architecture as *RLBot* but with the episodic and semantic memories running concurrently. The episodic memory captures episodes based on the event information in the working memory. An event from the UT game is encoded as a vector shown in Figure 7. There are four input fields in episodic memory for location, state, selected behavior, and the reward received. In the experiment, the vigilance of all input fields and the F_2 field are set to 1.0 and 0.9 respectively so that it tends to always store distinct events and episodes in response to the incoming events.

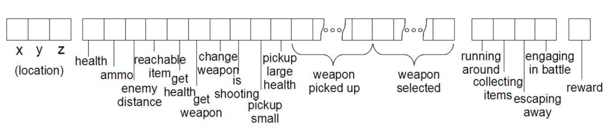


Fig. 7: Input vector to *RLBot* and *RLBot++*.

As described in Section V, the semantic network is applied to learn weapon effectiveness in the experiment. The network has three input fields: the Weapon field representing the identity of the weapon (F_1^a); the Distance field representing the distance between the agent and its opponent at the time of shooting (F_1^b); and the Effectiveness field representing the chance to kill the enemy (F_1^c). In the experiment, the vigilance of the Weapon (ρ^a), Distance (ρ^b), and Effectiveness (ρ^c) fields are 1.0, 0.9, and 0.8 respectively. The learning rate β^a, β^b, and β^c are 1.0, 0.1, and 0.2 respectively. Similar to the action selection process with procedural memory, the agent reasoning system can use the knowledge in the semantic memory by providing the current distance to the opponent while setting up the effectiveness to maximum (the greatest chance of killing) as memory cues. The retrieved values support the agent to decide which weapon to select during the battle. If the cue is not recognized, a random weapon is selected.

TABLE III: Sample semantic rules learnt in UT

```
IF distance is not so far [1800 2099]
  THEN ASSAULT_RIFLE effectiveness 0.07

IF distance is very near [300 599]
  THEN SHOCK_RIFLE effectiveness 0.946
```

Note: the largest visible distance to enemy is 300

Table III illustrates the sample learned rules of weapon effectiveness in symbolic forms. Each rule corresponds to a category node in F_2 layer of the semantic memory. The generalization employed using Fuzzy operators makes it possible to represent the values of antecedents with a range of values. Table III also shows the symbolic categorization of the distance range for interpreting the rules.

2) Results and Discussion: Experiments are conducted by letting *RLBot*, *RLBot++* and the memory-based *RLBot* (i.e. *MemBot*) to individually play against *AdvanceBot*. A single experiment run consists of 25 games or trials, which is ended whenever an agent kills or is killed by another agent.

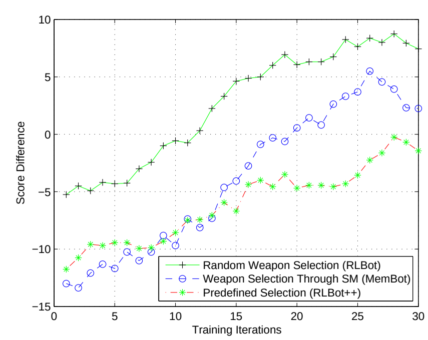


Fig. 8: Performance of *RLBot*, *RLBot++*, and *MemBot* over 25 trials.

Figure 8 shows the performance of both *RLBot*, *RLBot++* and *MemBot* in terms of game score differences against *AdvanceBot* averaged over four independent runs. From the performance plotting of *RLBot*, it dominates over its hard-coded opponent gradually. It shows that the procedural memory facilitates the agent through interacting with the environment and enhances its learning capability. By comparing its performance with *MemBot*, the experiment also confirms that the incorporation of the episodic and semantic memory modules further improves the learning which results in a much better performance than using the reinforcement learning alone (i.e. *RLBot*). This indicates that the semantic memory successfully learns useful knowledge for the weapon selection portion of the reasoning mechanism. The performance of *MemBot* can eventually reach the same level as the weapon selection optimized rules (i.e. *RLBot++*).

B. An Illustrative Domain: The Toads and Frogs Puzzle

The Toads and Frogs puzzle provides a benchmark domain, wherein reinforcement learning can be performed with a variety of semantic knowledge on the game domain and playing strategies. At the starting configuration, three toads are placed on the three leftmost squares of a seven-square board while three frogs are placed on the three rightmost squares (Figure 9). The central square is initially empty. The goal of the game is to switch the animal positions by having the toads occupy the three rightmost and the frogs occupy the three leftmost squares. A square can be occupied by only an animal at a time, and an animal can move only into the empty square. Toads can move only rightward and frogs only leftward. There are two possible types of move: a *Slide* to the next empty square and a *Jump* over an animal of a different type to an empty position with a distance of two squares away. The move cannot be retracted once it has been completed.

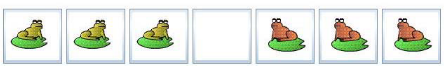


Fig. 9: The Toads and Frogs puzzle.

Along a solution path, some moves are forced as there is only a single feasible move (either a *Jump* or a *Slide* only) based on the current animal positions, while in the remaining cases, a decision has to be made to choose between two *Slides* (i.e. Slide-Slide Choice), or between a *Jump* and a *Slide* (i.e. Jump-Slide Choice), or between two *Jumps* (i.e. Jump-Jump Choice).

1) Semantic Memory: **Move Validity:** One straightforward form of semantic knowledge is the feasibility/legality of moving at a certain location based on the current status of the puzzle. According to the puzzle description, the moving feasibility depends on the contents (Toad/Frog/Space) of both current square in consideration and its nearby squares within the distance of two squares away. We summarize four symbolic rules on checking the moving feasibility in Table IV.

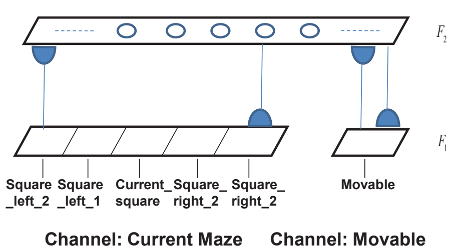


Fig. 10: Semantic knowledge on move validity.

In the listed rules, the attribute $current\_square$ indicates the content of the square in consideration, while $square\_left/right\_n$ ($n = 1, 2$) represents the content of the square n-square away on the left/right side of the current square. This type of semantic knowledge can be modeled as a fusion ART network as shown in Figure 10. The proposed network model has two F_1 fields: the Current_Maze input field consists of five attributes representing the current state of the considered and nearby squares, while the Movability field stores a single attribute value indicating the validity of moving in the current situation. Hence, to represent all the feasible

TABLE IV: Sample semantic rules on movability.

```
IF current_square is Toad, and square_left_1 is Space,
  THEN movable is true;

IF current_square is Toad, and square_left_1 is Frog,
     and square_left_2 is Space,
  THEN movable is true;
```

TABLE V: Sample semantic rules on JRND strategy.

```
IF MOVE1 is Jump, and MOVE2 is Jump,
  THEN PERFORM_MOVE1 is true, and PERFORM_MOVE2 is true;

IF MOVE1 is Jump, and MOVE2 is Slide,
  THEN PERFORM_MOVE1 is true, and PERFORM_MOVE2 is false;

IF MOVE2 is none,
  THEN PERFORM_MOVE1 is true, and PERFORM_MOVE2 is false;
```

moves based on a certain puzzle status, the network requires a total of seven memory cues, each of which represents one of the seven squares in the puzzle as the $current\_square$. The successful retrieval of any stored rule in the network indicates a valid move at the current square.

Jump and Random Strategy: As derived by a previous cognitive study on the Toads and Frogs puzzle [14], several strategics are known, which can be used to solve the puzzle. Each of these strategies can be modeled as a set of semantic memory networks. The Jump and Random strategy (JRND) states that the player should always perform the *Jump* for each Jump-Slide choice and choose a random action while facing a Slide-Slide choice. A random action should be picked for the Jump-Jump choice, since the Jump-Jump situation always leads to a dead end eventually. The JRND strategy can be expressed using the semantic rules presented in Table V. From Table V, JRND strategy also requires additional knowledge on the feasible move(s) based on each possible puzzle state. Some examples of such semantic knowledge are given in Table VI.

Figure 11 illustrates how a set of fusion ART networks may interact with each other to implement the JRND strategy. The JRND strategy involves two semantic networks: a network on the types of valid move(s) based on each possible puzzle status (Figure 11-a) and a network to implement JRND (Figure 11-b). At each step of the puzzle solution, the network on the types of feasible move(s) is firstly retrieved for the set of feasible moves and their types based on the current puzzle status. This retrieved information is shared with the JRND semantic network through the common working memory to recommend a move based on the JRND strategy.

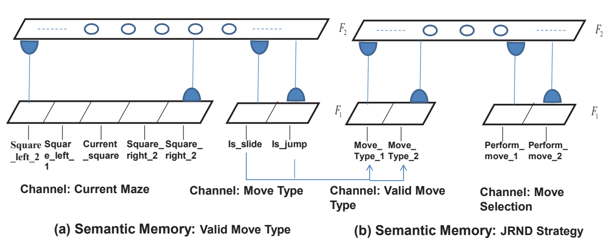


Fig. 11: Semantic memory and interactions on jump and random strategy: at each decision cycle of the puzzle play, network (b) determines the right move to take based on the set of valid moves determined by network (a).

TABLE VI: Sample semantic rules on move type.

```
IF current_square is Toad, and square_left_1 is Space,
  THEN is_slide is true;

IF current_square is Toad, and square_left_1 is Frog,
     and square_left_2 is Space,
  THEN is_jump is true;
```

2) Procedural Memory: The procedural memory learns and performs the move selection through the reinforcement learning algorithm as stated in Algorithm 2. The state, action, and reward vectors in Figure 12 correspond to the input fields in a multi-channel ART network. Move pattern in the state vector represents the move currently selected. The action vector indicates the next move to be selected. Based on the state field and the reward, the network searches the best match category node and reads out the output to the action field indicating the move type to be selected. The network then receives feedbacks in terms of the new state and any reward given by the environment.

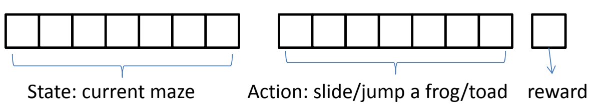
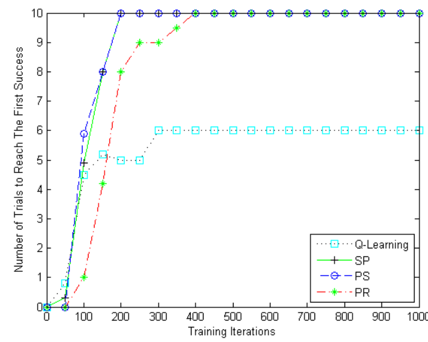


Fig. 12: The state, action and reward representation for Toads and Frogs puzzle.

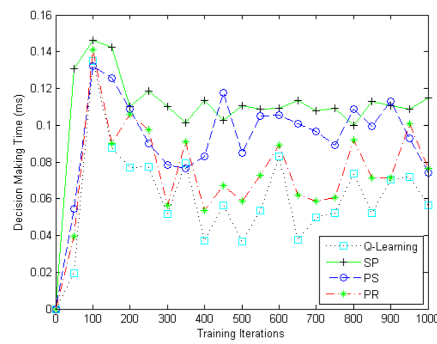
The network learns by updating the weighted connections according to the feedback received and applying temporal difference methods as described by Algorithm 2 to update the reward field. The agent receives the reward signal (1 or 0) whenever it succeeds or fails in resolving the puzzle at one trial. The immediate rewards will also be given after each move based on its improvement on distance from the resultant puzzle state to the desired final one. In this way, procedural knowledge is continually learned and acquired on the move selection while solving the puzzle.

3) Results and Discussion: Experiments are conducted by playing the Toads and Frogs puzzle game using different types of memory and their combinations. A single experiment run consists of 1000 games trials. The performance is measured and compared by the success rates and the number of trials to reach the first successful game, averaging over ten independent experiment runs.

Table VII shows the performance of five different experiment configurations embedded with various memory options, including (1) the pure procedural memory learning (based on TD-FALCON [25]); (2) the pure procedural memory learning using the standard Q-learning method [5]; (3) semantic knowledge on feasible moves on each moving step only; (4) semantic memory modeling of Jump and Random (JRND) strategy; (5) procedural memory learning combined with the semantic knowledge on feasible moves via SP interactions; and (6) procedural memory learning combined with the semantic knowledge on feasible moves via PS interaction. From Table VII, the JRND strategy and semantic knowledge on feasible moves both produce success rate of less than 50%. This poor performance shows that these two configurations contain insufficient knowledge (both semantic and procedural)



(a) Task Performance



(b) Decision-Making Time

Fig. 13: Performance comparison on Toads and Frogs puzzle among different types of memory interactions.

required to solve the Toads and Frogs puzzle. The experiment also shows that the fusion ART-based procedural memory model is able to achieve a better performance compared with the standard Q-learning method. By further comparing the performance of pure procedural learning with its combinations with semantic memory, the experiment confirms that the interaction of the procedural and semantic memory modules further improves the learning resulting in a higher level of performance.

We further investigate and compare the performance of the agents using different models of procedural-semantic memory interactions (i.e. memory options (1), (2), (5) and (6)). Figure 13(a) shows the agents' task performance in terms of the number of successful trials achieved in the last ten game trials over time. It is evidenced that both ways of memory interactions (shown as SP and PS) help to solve the puzzle within a smaller number of trials, compared with pure procedural learning (shown as Procedural). Figure 13(b) shows the learning efficiency in terms of the execution time taken for each decision-making. It can be seen that PS interaction has a shorter decision making time compared to that of SP interaction, due to the targeted searching approach to reduce the frequency of memory access as well as decision time. Compared with Q-learning, our procedural learning produces a higher success rate with the cost of a slightly longer decision making time.

C. Empirical Study on StarCraft

1) Semantic Memory In StarCraft: Resource gathering, building construction and unit production are the three main

TABLE VII: Performance of various methods, based on pure procedural, pure semantic and combined memory, on Toads and Frogs puzzle.

	Success Rate (%)	First Successful Trial
Pure Procedural Learning (TD-FALCON)	87.62	124.1
Pure Procedural Learning (Q Learning)	55.53	171.1
Feasibility Check	0.49	357.4
JRND	25.45	3.4
Procedural Learning with Feasibility Check (SP)	93.41	65.9
Procedural Learning with Feasibility Check (PS)	94.02	59.1

TABLE VIII: Sample semantic rules on resource conditions.

```
IF mineral is [50, +∞ )
   and gas is [25, +∞ )
  THEN HAVE_ENOUGH_RESOURCE_TO_BUILD_FIREBAT is true

IF mineral is [50, +∞ )
   and gas is [100, +∞ )
  THEN HAVE_ENOUGH_RESOURCE_TO_BUILD_STARPORT is true
```

tasks in a StarCraft game. Each construction or production activity consumes a certain level of resources such as mineral, gas and supply. Each attempt of building activity without sufficient resources will be denied by the game environment. Due to this dependency among the three tasks, semantic knowledge on the resource condition for building construction and unit production is critical to develop a fast and efficient territory building-up during the early stage of the StarCraft game. In the semantic memory module, resource condition can be expressed as the causal relation rules between resource level and its feasible action(s). The semantic memory is in turn encoded based on the fusion ART model as shown in Figure 14. The proposed SM model has two input (F_1) fields: the Resource field represents the resource level in terms of mineral, gas and supply count, while the Action field represents the various construction or production actions. Each node in F_2 represents a causal relation rule. Some examples of the causal rules are given in Table VIII.

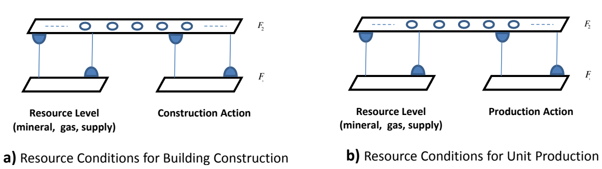


Fig. 14: Representation of semantic knowledge on resource conditions.

2) Empirical Comparison: To investigate how the individual memory modules contribute to the overall performance, learning agents embedded with different memory modules are tested and compared. We evaluate three types of memory-based agents, namely the PR agent with only the procedural memory, the SP agent incorporating the multi-memory system using the SP interaction, and the PS agent using the PS interaction. The semantic knowledge employed is the resource conditions for both building construction and unit production presented previously.

Experiments are conducted by letting the PR agent, the SP agent and the PS agent to individually play in the StarCraft game. We compare the performance of different agents in terms of the numbers of units, buildings and resources constructed or collected, at the end of each trial/game. Figure 15 shows the empirical comparison among the three configurations organized in terms of resource management, building construction and unit production. It also provides the overall performance of the three configurations, in terms of overall game score and average decision time. The overall game score is computed as the weighted sum of unit, building and resource counts at the end of game and normalized between 0 and 1. The plots are obtained by averaging over 20 experiment runs, of which each consists of 200 games trials.

As shown in Figure 15(a), the SP agent and PS agent have provided a better performance in the task of resource management, comparing with that of the PR agent. The use of semantic memory has helped the agents to collect the right amount of resource. The PS agent has a slightly better performance compared with the SP agent, due to the targeted searching of semantic memory, which reduces the excessive knowledge access and thus decision time.

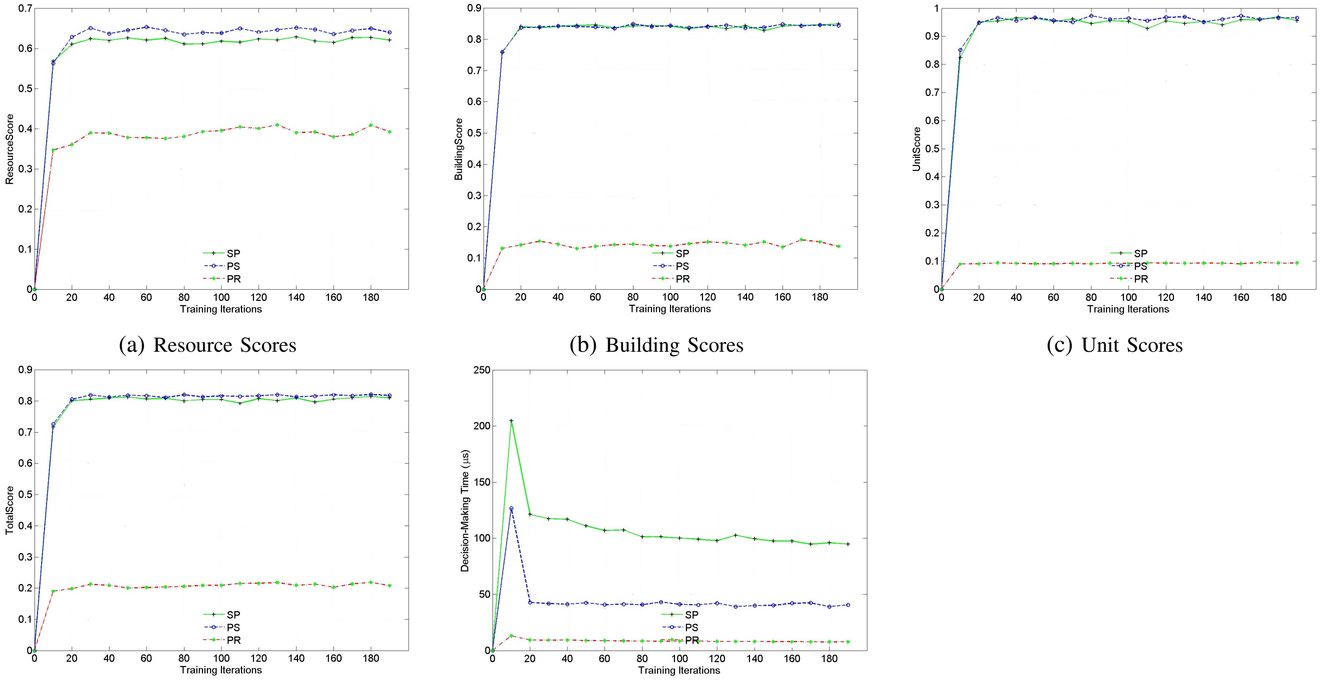
Note that the unit production tasks usually have a high dependency on building construction, as the production of most of the units requires some buildings as a pre-condition. As shown in Figure 15(b), both the SP and PS interactions have helped to produce more units across various types, compared with that using pure procedural learning.

From Figure 15(c), both the SP and PS interactions provide a better performance on building construction, compared with pure procedural learning. In Starcraft, building construction is usually more expensive and time-consuming, and hence is critical to achieving the overall game target. The high efficiency of this task learning therefore leads to more successful completion of building construction. Combined with the results for building construction, the SP and PS interactions have led to faster building construction, facilitating the unit production task.

As shown in Figure 15(d) and (e), both the SP and PS interactions also lead to a better overall performance during the game. In terms of response time, the PS interaction produces a shorter decision cycle due to its targeted memory search. At the starting stage of the game, both the SP agent and PS agent incur a longer decision cycle as procedural memory learning heavily depends on the retrievals from semantic memory. However, the decision time declines as the knowledge is gradually transferred from semantic to procedural memory.

IX. Conclusion

This paper has presented a novel computational approach to the modelling of semantic memory based on self-organizing neural networks. Using a unified set of computational principles and algorithms, we show how various forms of semantic



(a) Resource Scores (b) Building Scores (c) Unit Scores

(d) Total Scores (e) Decision-Making Time

Fig. 15: Performance comparison between learning agents using SP interaction, PS interaction and pure procedural learning (PR).

knowledge can be encoded and learned in a unified and consistent manner. We have conducted empirical experimental evaluation on the proposed semantic memory model using the fast-pace first-person shooting Unreal Tournament game, the Toads and Frogs puzzle, as well as the Starcraft strategy game. The experimental results show that the model is able to learn/encode various types of semantic knowledge across different domains. It also indicates that the semantic memory model is able to enhance the agents' performance consistently in various decision making tasks, through its collaborative interactions and knowledge transfer with other memory components.

In our current work, the proposed memory consolidation process requires the prior knowledge on the specific types of semantic knowledge to learn. Consequently, the effective learning of semantic memory requires the involvement and judgement of the domain experts for specific applications. Also, we are currently focusing on the three key types of semantic knowledge studied. Although it is general enough to apply to a wide range of domain, this prevents the discovery of novel knowledge types and may limit the applicability of the proposed framework. Going forward, one important extension of our work is therefore to identify additional possibly more complex semantic representations important in real-life domains and to extend our framework for representing and learning a more generic semantic structure.

ACKNOWLEDGEMENT

This work is supported in part by the DSO National Laboratories under Research Grant DSOCL11258 and a research scholarship from the School of Computer Engineering, Nanyang Technological University.

REFERENCES

[1] J. R. Anderson. *Rules of the mind*. Lawrence Erlbaum Associates, Hillsdale, 1993.

[2] J. Binder and R. Desai. The neurobiology of semantic memory. *Trends in Cognitive Science*, 15(11):527–536, 2012.

[3] G. A. Carpenter and S. Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 37:54–115, June 1987.

[4] A. M. Collins and M. R. Quillian. Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior*, 8(2):240–247, 1969.

[5] T. Eden, A. Knittel, and R. Uffelen. Reinforcement learning. Web: http://www.cse.unsw.edu.au/ cs9417ml/RL1/index.html.

[6] M. J. Farah and J. L. McClelland. A computational model of semantic memory impairment: Modality specificity and emergent category specificity. *Journal of Experimental Psychology: General*, 120(4):339–357, 1991.

[7] M. A. Gluck and C. E. Myers. Hippocampal mediation of stimulus representation: A computational theory. *Hippocampus*, 3(4):491–516, 1993.

[8] T. L. Griffiths, M. Steyvers, and J. B. Tenenbaum. Topics in semantic representation. *Psychological Review*, 114(2):211–244, 2007.

[9] G. E. Hinton. Implementing semantic networks in parallel hardware. In G. E. Hinton and J. A. Anderson, editors, *Parallel Models of Associative Memory*, pages 161–187. Lawrence Erlbaum Associates, Hillsdale, 1981.

[10] M. W. Howard, K. H. Shankar, and U. K. Jagadisan. Constructing semantic representations from a gradually changing representation of temporal context. *Topics in Cognitive Science*, e(2):48–73, 2011.

[11] B. Kosko. Fuzzy cognitive maps. *International Journal of Man-Machine Studies*, 24:65–76, 1986.

[12] J. E. Laird and S. Mohan. A case study of knowledge integration across multiple memories in soar. *Biologically Inspired Cognitive Architectures*, 8:93–99, 2014.

[13] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning memory. *Psychological Review*, 102(3):419–457, 1995.

[14] F. Missier and D. Fum. Declarative and procedural strategies in problem solving: Evidence from the toads and frogs puzzle. In *Proceedings of the Twenty-Fourth Annual Conference of the Cognitive Science Society*, pages 262–267, 2002.

[15] S. T. Mueller and R. M. Shiffrin. REM-II: a model of the development co-evolution of episodic memory and semantic knowledge. In

Proceedings of International Conference on Development and Learning, volume 5, 2006.

[16] A. Oltramari and C. Lebiere. Extending cognitive architectures with semantic resources. In *Proceedings of the Fourth Conference on Artificial General Intelligence (AGI 2011)*, 2011.

[17] R. C. OReilly, R. Bhattacharyya, M. D. Howard, and N. Ketz. Complementary learning systems. *Cognitive Science*, 38:1229–1248, 2014.

[18] M. A. L. Ralph, C. Lowe, and T. T. Rogers. Neural basis of category-specific semantic deficits for living things: evidence from semantic dementia, hsve and a neural network model. *Brain*, 130:1127–1137, 2007.

[19] G. Recchia, M. Jones, M. Sahlgren, and P. Kanerva. Encoding sequential information in vector space models of semantics: Comparing holographic reduced representation and random permutation. In *Proceedings of the 32nd Annual Cognitive Science Society*, pages 877–882, 2010.

[20] G. J. Rinkus. A neural model of episodic and semantic spatiotemporal memory. In *Proceedings of the 26th Annual Conference of Cognitive Science Society*, pages 1155–1160, Chicago, 2004. LEA.

[21] D. E. Rumelhart. Brain style computation: learning and generalization. In *An introduction to neural and electronic networks*. Academic Press Professional, San Diego, 1990.

[22] R. Sun. Memory systems within a cognitive architecture. *New Ideas in Psychology*, 30:227–240, 2012.

[23] R. Sun and R. Mathews. Implicit cognition, emotion, and meta-cognitive control. *Mind and Society, the special issue on Dual Processes Theories of Language and Thinking*, 11:107–119, 2012.

[24] A.-H. Tan, G. A. Carpenter, and S. Grossberg. Intelligence Through Interaction: Towards A Unified Theory for Learning. In *International Symposium on Neural Networks (ISNN) 2007*, volume 4491, pages 1098–1107, Nanjing, China, June 2007. LNCS.

[25] A.-H. Tan, N. Lu, and X. Dan. Integrating Temporal Difference Methods and Self-Organizing Neural Networks for Reinforcement Learning with Delayed Evaluative Feedback. *IEEE Transactions on Neural Networks*, 19(2):230–244, February 2008.

[26] H. S. Terrace and J. Metcalfe. *The Missing Link in Cognition: Origins of Self-reflective Consciousness*. Oxford University Press, 2005.

[27] E. Tulving. *Elements of episodic memory*. Oxford University Press, 1983.

[28] W. L. Tung and C. Quek. eFSM: A Novel Online Neural-Fuzzy Semantic Memory Model. *IEEE Transactions on Neural Networks*, 21(1):136–157, 2010.

[29] M. Ullman. The declarative/procedural model of language. In *Encyclopedia of the Mind*, pages 224–226. Sage Publications, 2013.

[30] W. Wang, B. Subagdja, A.-H. Tan, and A. Z. Starzyk. Neural modeling of episodic memory: Encoding, retrieval, and forgetting. *IEEE Transactions on Neural Networks and Learning Systems*, 23(10):1574 – 1586, 2012.

[31] M. A. Wilson and B. L. McNaughton. Reactivation of hippocampal ensemble memories during sleep. *Science*, 265:676–679, 1994.

[32] D. Xiao and A.-H. Tan. Self-organizing neural architectures and cooperative learning in multi-agent environment. *IEEE Transactions on Systems, Man and Cybernetics - Part B*, 37(6):1567–1580, 2007.

[33] M. Yassa and Z. Reagh. Competitive trace theory: A role for the hippocampus in contextual interference during retrieval. *Frontiers in Behavioral Neuroscience*, 7:107, 2013.