Neural modeling of multiple memory systems and learning

Wang, Wenwen

2015

Wang, W. (2015). Neural modeling of multiple memory systems and learning. Doctoral thesis, Nanyang Technological University, Singapore.

https://hdl.handle.net/10356/62219

https://doi.org/10.32657/10356/62219

NANYANG TECHNOLOGICAL UNIVERSITY

Neural Modeling of Multiple Memory Systems and Learning

A Thesis Submitted to
the School of Computer Engineering
of Nanyang Technological University

by

Wang Wenwen

in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy of
Computer Engineering

February 2015

Abstract

Memory forms the basis of human's intelligence enabling us to reason and act appropriately based on past experiences. According to the theory of multiple memory systems, there are two major types of long term memories, namely declarative memory and procedural memory. Whereas procedural memory refers to the implicit knowledge of performing tasks, declarative memory can be further divided into two related components, namely episodic memory of specific past experience and semantic memory of concepts and general facts.

In view of the importance of memory in developing intelligent capabilities, this research aims to study the neuropsychological principles and constraints of the memory systems in the brain and develop a computational model of the multiple memory systems. Furthermore, this research aims to embed these memory modules into autonomous agents so as to improve their decision making and problem solving capability.

To this end, this thesis presents a biologically inspired computational model of multiple memory systems which are able to learn and co-evolve, in response to a continuous stream of sensory input and feedback from the environment. Using self-organizing neural networks as the building block, the multi-memory architecture presents a neurally-plausible model for each type of the long-term memory systems, including episodic memory, semantic memory and procedural memory. The model further encompasses a set of emergent processes, through which the various memory modules may transfer knowledge

and cooperate with each other, for decision making and problem solving without the use of a centralized executive control.

Specifically, as part of the multi-memory architecture, the episodic memory module, based on an extension of fusion Adaptive Resonance Theory (fusion ART) network, extracts key events and encodes their spatio-temporal relations by creating cognitive nodes dynamically. The episodic memory, called EM-ART, further incorporates a novel memory search procedure, which performs parallel search of stored episodic traces continuously. We present experimental studies, where EM-ART is evaluated based on the encoding efficiency and recall accuracy. Our experimental results show that EM-ART produces more robust performance in encoding and recalling events and episodes with incomplete and noisy cues, compared with prior models of spatio-temporal memory.

Similarly, based on fusion ART, the semantic memory module presents a unified set of representation and learning methods for various types of semantic knowledge. A general procedure of memory consolidation is also proposed and implemented, wherein episodic memory is consolidated and transferred to the more permanent semantic memory. The declarative memory model, consisting of episodic and semantic memory, has been embedded into a reinforcement learning agent in a game environment called Unreal Tournament, wherein the declarative memory continuously acquires knowledge about the environment. The experiments show that memory consolidation is able to extract useful knowledge to enhance the performance of high level cognitive tasks.

Finally, based on a specialization of three-channel fusion ART, the procedural memory module acquires the procedural knowledge and skills through interacting with the environment via reinforcement learning. More importantly, we formalize two major types of memory interaction, wherein the procedural memory and the semantic memory cooperate in decision making and problem solving. We investigate the overall performance of the declarative-procedural memory system embedded into autonomous learning agents in

two problem domains: (1) the Toad and Frog puzzle and (2) a strategic game known as Starcraft Broodwar. The results show that the cooperative interaction between declarative knowledge and procedural skills can lead to a significant improvement in both learning efficiency and performance of the learning agents.

Acknowledgments

This thesis would not have been possible without the guidance and the help of several individuals who kindly contributed their valuable assistance in the preparation and completion of this study.

First and foremost I wish to render my heartfelt appreciation to my advisor, Associate Professor Ah-Hwee Tan, who has supported me throughout this project with his patience and knowledge. Without his consistent support and supervision, the research would not have been so enriching and fulfilling.

This work would also have not been finished without the contribution from Dr Budhitama Subagdja, who works as a research fellow in my research group. Thanks for his great advises and contributions to make this research more complete and mature.

I gratefully acknowledge Professor Janusz A. Starzyk for his kind guidance, advice, and crucial contribution to this research. During his visiting in our school, his involvement in this research inspires my creativity and enthusiasm on my research, which benefits me for a long time.

Last but not least, I would like to thank the School of Computer Engineering and the Defence Science Organization (DSO) National Laboratories for the research scholarship. Many thanks also go to the staffs at the C2I lab for their technical support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Memory is a critical component of the brain for reasoning and decision making. It forms our understanding of the environment and affects our daily behaviors by providing past relevant experiences to extend momentary perceptual range, enable goal tracking, support planning, and improve learnt knowledge [1]. The long-term memory in the brain has been well recognized as multiple memory systems [9, 10] consisting of notably declarative memory and procedural memory. Declarative memory is an explicit record of what we encounter and what we learn [11, 12]. Procedural memory, on the other hand, refers to the implicit memory of skills and reflex responses, wherein the knowledge is usually difficult to articulate or explain. While procedural learning is essential to the development and utilization of both motor and cognitive skills, the declarative memory represents the high level concepts and knowledge which forms the basis of our understanding and guides us in reasoning and decision-making.

As an integral component of our long-term memory, declarative memory has been known in psychology as a type of memory enabling one to consciously and deliberately remember experiences or knowledge learnt from the past. Typically, declarative memory is further divided into episodic memory and semantic memory. While episodic memory enables one to remember personal experiences that can be explicitly stated, semantic memory stores meanings, concepts, rules, and general facts unrelated to specific experiences [12].

As a part of the declarative memory, episodic memory has been identified in recent research 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 [13]. Additional research on the role of episodic memory and hippocampus (an area of the brain believed to be associated with episodic memory) in animals also indicates that episodic memory is an important part of an individual to learn about context and configurations of stimuli. In particular, Morgan and Squire have shown that during reinforcement learning tasks, hippocampus is critical for representing relationships between stimuli independent of their associations with reinforcement [14]. The specific functionalities mentioned above suggest that episodic memory should not be just a storage of one's past experiences, but should also support the representation of complex conceptual and spatio-temporal relations among one's experienced events and situations. Therefore, a major concern of modeling episodic memory is on how it can be used as an effective storage with flexible retrieval mechanism of those past experiences and their spatio-temporal relations. Furthermore, a dynamic memory management is also required to prevent the ever-growing memory size as the model continuously records daily experiences.

On the other hand, semantic memory represents high level concepts and knowledge extracted from specific experiences without explicit referencing. Various kinds of general knowledge are situated at different levels of cognitive hierarchy to influence the decision making based on experiences, including Is-A (i.e. concept) relation, association relation, and causal relation. Hence, a study on semantic memory requires a unified set of representation and learning methods for various types of knowledge and concepts. Moreover, other key issues in studying semantic memory include: (1) what and when experiences can be selected to form semantic concepts and knowledge; and (2) how these knowledge can be used to support the related cognitive functions and task performance.

Known as the counterpart of declarative memory, procedural memory is the memory on how to perform certain kinds of action or behaviors. It learns well-defined action sequences and execute action-related procedures under the level of consciousness. Through intensive rehearsal on complex action sequences from one's own experiences, procedural memory guides the association of all relevant cognitive modules to accomplish various tasks in hand, and hence serves a critical role for the further development of cognitive and motor skills. Therefore, for a proper model of procedural memory, our research focuses on studying the crucial issues on: (1) how the past experiences can be encoded, learnt and retrieved in a unconsciously and automatic manner; and (2) how the ever-growing experiences can affect the procedural knowledge previously learnt and hence lead to potential skill improvements.

Although these three types of long-term memory represent distinct knowledge and support different cognitive functionalities, prior research have suggested that they interact intensively with one another to serve their roles and functionalities and affect our learning and behavior. Besides the facts that most skills in the real life is combined with both procedural and declarative components, there is a common belief that declarative memory initiates and provides the foundation for procedural learning [15]. Previous research [16] on neuroscience also indicates that declarative memory is also directly associated with the acquisition and subsequential memory (re)consolidation of procedural knowledge.

As two complementary components of declarative memory, semantic and episodic memory also interact and influence each other regularly [17, 18]. On one hand, semantic memory can be considered as the outcome of the gradual knowledge transfer from episodic memory. The process of knowledge transfer forms semantic knowledge by extracting the set of common features among similar episodic experiences while removing the sensitivity to temporal-spatial context and the associations with specific experiences. On the other hand, semantic memory influences our daily activities in understanding as well as interacting with the environment, hence guides the formation of new episodic memory.

Figure 1.1 gives an overview of interactions among the different types of long-term memory, based on the neuroscience observations from [16, 15, 17, 18]. In this research, we shall study the individual structures and processes of memory encoding, learning and retrieving of the episodic, semantic, and procedural memories under the complex context of an integrated architecture. Specifically, the investigation will be conducted on how the different memories interact to support and enhance each other, in contrast to isolated studies on individual memories.

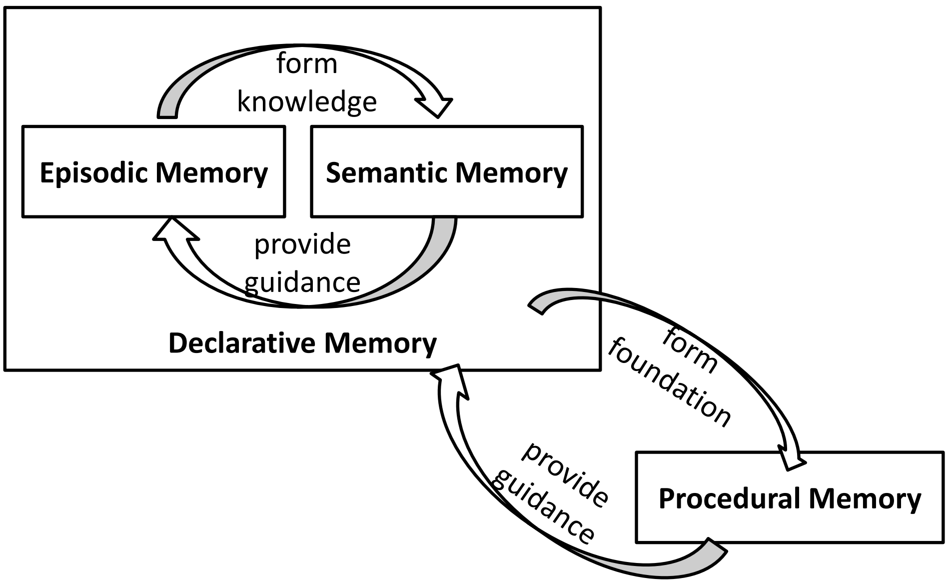


Figure 1.1: Overview of interactions among different types of long-term memory

1.2 Issue and Challenges

This section discusses the major issues and challenges on modeling multiple memory systems and their interactions. Section 1.2.1, section 1.2.2 and section 1.2.3 discuss the main issues of modeling each individual component of the long-term memory systems, namely episodic memory, semantic memory and procedural memory. Section 1.2.4 and section 1.2.5 highlight the major challenges for modeling the interaction among the various types of memories.

1.2.1 Modeling of Episodic Memory

As discussed in Section 1.1, two basic elements of episodic memory are events and episodes: An event can be described as a snapshot of one's experience, containing the "what', "when" and "where" information about the corresponding experience; while a temporal sequence of events that one experiences can be called an episode.

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 episodic novelty detection is to build an efficient storage matching scheme that can distinguish important entries from irrelevant ones across time based on some criteria that may be inexact or ambiguous. On one hand, the novelty detection should be sufficiently strict to distinguish highly similar but semantically different events; On the other hand, it should also be loose enough to tolerate minor differences between event sequence within an episode. Hence, the critical characteristics for the novelty detection scheme is its high efficiency in determining the significant differences between events/episodes.

Many existing episodic memory models have attempted to address some of these challenges. However, while some existing episodic memory models focus on the encoding the spatiotemporal relations among events, most still have limitations in capturing complex concepts and situations, including representing multiple-channel event and dealing with noisy inputs and imperfect information (e.g. [19, 1]). On the other hand, those models supporting the intricate relations of concepts and events are not able to process complex sequences of events as a whole (e.g. [5, 20, 2]).

Besides simply encoding the experienced events and their temporal associations, the episodic memory model should be able to explore and evaluate the importance of stored events and episodes in order to facilitate the process of decision making and reasoning. Major factors concerning the importance of event and episode at the moment of memory formation include rewards, prediction surprises, emotions, accessing frequency and forgetting. In addition, events, episodes and their importance may change in time as a result of interpretation.

The importance-based dynamic management of episodic memory should serve to strengthen and preserve important episodes and remove unimportant ones. Assuming the importance of the learned event and episode pattern is simply based on the frequency of activations, a dynamic management scheme can simply prune less important episodes on a periodic basis if their level of importance falls below a predefined threshold. One of the existing episodic memory models [2] has addressed the issue of dynamic memory. It has indicated the use of sleep for knowledge consolidation and active forgetting of less useful events. However, the implementation details are not provided.

In summary, the main elements of an episode within episodic memory include the following information as suggested by [13]:

- Individual events encoded as an aggregation of objects, concepts, actions and locations in a memorable scene
- Temporal relations between events as sequences of event features, rather than explicit time reference

- Dynamical management of event and episode importance based on rewards, surprises, emotions, interpretation, access frequency and aging
- Association between events (possibly across different episodes) to facilitate multi-episode recalling

1.2.2 Modeling of Semantic Memory

As mentioned previously, semantic memory refers to the memory of meanings, understandings, and other concept-based knowledge unrelated to specific experiences. It is the conscious recollection of the factual information and the general knowledge about the world [21]. In literature, researchers have studied many types of the semantic knowledge, called concepts, situated at different levels in cognitive hierarchy, including Is-A relation (e.g."Lily *is a* girl"), concept association (e.g. "weekend *is usually* happy and relaxing"), and causal relation (i.e. a "IF-THEN" rule) [11]. Therefore, to model semantic memory efficiently, this research requires general principles and processes for encoding and learning various types of semantic memory in a unified manner for all possible types of related semantic knowledge.

Although many semantic memory models have been proposed, most of them are limited to studying some specific aspects of the semantic memory: most early abstract models (e.g. [22] and [23]) focus on representation but not on learning of such semantic knowledge; In contrast, typical statistical models (e.g. [24], [25] and [19]) allow learning but only work for a limited form of semantic memory, namely correlation between words and concepts. As the more recent development of semantic memory modelling, some connectionist models (e.g. [26, 3]) support complex concept representation based on multi-modal sensory inputs, while others (e.g. [27, 28]) employ the single-modal representations to explore the correlations among semantic concepts. However, to our best knowledge, none of these existing work study the learning of the wide range of semantic relations using the multi-facet concept representation.

This research aims to develop a semantic memory model based on a unified set of computing principles, that encompasses multiple key functions of semantic memory, including concept formation, causal relation learning, and concept association. In designing the memory models and learning algorithms, we consider the principles and requirements summarized below.

- What should be the representation of semantic memory? How do we acquire such semantic memory? In fact, there are many types of semantic knowledge. It is important for a model to represent and learn multiple types of knowledge, including concept association, Is-A relation, and other types of semantic relations.

- Concepts are learned based on multi-modal sensory inputs. For example, a concept *"Chair"* has a multi-facet representation, including visual, auditory, and verbal. A model for concept formation should be able to learn concepts based on multi-modal sensory representation. Furthermore, remembering a concept in multiple representations provides the needed redundancy in concept recognition. The fusion of the multi-modal sensory representation at conceptual level also provides the capabilities in cross-modal association and retrieval.

- How do different types of semantic memory work together? The various semantic memory modules in the architecture should interact and function cooperatively.

- While most prior models deal with a concept or concept pair at a time, the human's working memory is capable of holding multiple concepts for further processing at one time [29]. The architecture should provide one or more working memory buffers for multiple concepts to be active at the same time.

1.2.3 Modeling of Procedural Memory

Procedural memory refers to the memory of how to perform actions or tasks, such as riding a bicycle or driving a car. In contrast to declarative memory, most of knowledge in procedural memory is verbally inexpressible, automatically learned and acquired below the level of conscious awareness [30].

Existing computational models of procedural memory can be classified into three categories, namely connectionist models, fragment-based models and the hybrid models. The connectionist models of procedural memory (e.g. [31]) usually learn the complex procedural procedures as temporal-ordered sequences of situation-action association pairs. On the other hand, fragment-based or chunking models (e.g. [32]) acquire procedural knowledge through a case-based learning of memory chunks or fragments. These models records the situation-action pairs without the explicit recording of action sequences. More recently, the hybrid models of the two procedural learning paradigms further investigate how the cognitive skills and capabilities can be supported by the collaborative interaction between procedural and declarative learning. In these models, procedural memory (e.g. [33]) controls and executes various real-time decision making based on the sub-symbolic procedural knowledge representation, while the symbolic and explicit rules are gradually learnt as the declarative knowledge.

Although the existing models on procedural memory have been applied and investigated in a wide range of practical domains, there are two key issues on procedural learning which have not been well addressed from these works. These considerations serves as our guiding principles as highlighted below.

- Procedural memory learning should consist of a complex process of priming, wherein the continuously-growing experience keep shaping the memory to influence and guide the behavior in future. Through priming, the memory retrieval on procedural knowledge can be more than repeating a specific instance from past experience, which has been typically shown in the chunking models.

- Although the hybrid models on procedural memory have been focusing on understanding how the procedural knowledge can be translated into the corresponding declarative representation, there is a need to develop an unified model of procedural-declarative memory interactions across various tasks and domains.

1.2.4 Modeling Interaction within Declarative Memory

As discussed previously, within the declarative memory systems, semantic memory is formed based on high level concepts and knowledge, while episodic memory is the collection of low level instances [12]. Under certain circumstances, semantic knowledge could be created from several episodic experience and the abstract semantic memory concepts sometimes could be traced back to specific episodic memory instances [17]. During the modeling of the interaction between episodic and semantic memory, the following issues and concerns are identified:

- What are the contents from episodic memory copied to semantic memory for abstraction of semantic knowledge? Generally speaking, episodic memory captures a snapshot for each encounter incurred previously which may involve large numbers of trivial details. To model the interaction from episodic to semantic memory, this research requires an algorithm to select relevant inputs or attributes from the stored event patterns as well as their spatio-temporal relations in order to extract useful semantic knowledge.

- How to perform knowledge transfer from episodic to semantic memory? Since semantic memory refers to various types of semantic knowledge, this research requires a general set of principles and procedures to consolidate the learned events and episodes from episodic memory to various forms of semantic knowledge in a unified way.

- When does knowledge transfer take place? Knowledge transfer can be time consuming due to the massive amount of information stored in episodic memory. The timing of transfer should be selected in a way that the transfer process will not affect the performance of other functionalities.

1.2.5 Modeling Interaction between Declarative and Procedural Memory

In this work, we also aim to develop an interaction model between declarative memory and procedural memory to facilitate the overall process of decision making and problem solving. For the design of the interaction model, several considerations serve as our guiding principles are highlighted below.

- The interaction model should contain at least two types of long-term memory modules, including the semantic memory of meanings, concepts, rules, and general facts and the procedural memory for performing actions based on direct state-action pairings.

- The interaction model should enable a wide variety of interactions between the two memory modules. The interaction should preferably be achieved via the neural pathways connecting the different memory modules, such that the knowledge and information from the different memory modules can be shared and commonly used by all. This dual-memory systems should also model the knowledge transfer processes between different memory systems especially from semantic to procedural memory.

- The interaction model should allow the individual and parallel processes of different memory modules running in varying time scales. In the dual-memory systems, semantic and procedural memory modules should run online in a concurrent manner. Each learning process should individually provide a fast and efficient learning even when the other complementary module(s) fail.

- The interaction model should enable the emergence of intelligence from the various types of memory interactions. The interactive declarative-procedural memory models should contribute to a better decision making by utilizing the information from the entire knowledge base. During this process of decision making with dual memory interactions, each individual model may also require subsequent learning on the given situation with the additional information shared from other types of memories.

- As information and knowledge is shared from different memory modules, the decision making process should resolve the potential information conflicts and errors.

- The interaction model should suppport the efficient learning and fast responses as required by the online task domains. As various memory modules run concurrently, each module should maintain the efficient performance for its own operation, such as rule/memory trace retrieval, encoding, and deletion.

1.3 Research Objectives

This research aims to develop a computational model of the multiple memory systems in the brains, which can be subsequently incorporated into brain-inspired cognitive architectures. Through representing, learning, and processing of different types of memory, this research discusses and evaluates the possibility of acquiring and co-evolving both declarative and procedural memory to enhance complex cognitive functionalities and integrated performance.

As part of the multi-memory architecture, the incorporated episodic memory shall support the learning of all possible complex situations as well as their spatio-temporal relations [12]. In respond to a continual stream of sensory input and feedback received from the environment, the episodic memory module should be able to capture and explore details of all the encounters with the consideration of their spatio-temporal ordering through the form of one-shot learning on events and episodes. To achieve an efficient representation and learning of events and episodes, the episodic memory module requires a well-defined generalization and matching scheme that distinguishes significant differences among different events and episodes while tolerating minor changes within a learned pattern. As the sensory and feedback signals from real-world environment are supposed to be noisy, incomplete, redundant or even conflicting, this study on episodic memory modeling should also investigate the robust learning of events and episodes dealing with imperfect information. Moreover, besides fast and accurate memory encoding and retrieving, this research takes further consideration and emphasizes on an efficient dynamic management of learned events and episodes. Due to the real-time learning of episodic memory, to prevent an ever-growing memory size, this research holds a view such that storage in episodic memory is not permanent. Dynamic management of memory should be incorporated into the episodic memory model in a way that it only holds the storage of important experiences while discarding the insignificant information.

As introduced previously, semantic memory represents different types of concepts and knowledge sitting at various levels of knowledge hierarchy as defined in [11]. Therefore, the research on the modeling of semantic memory requires a unified set of algorithms and procedures for encoding and retrieving many possible types of semantic knowledge on concepts, associations and causal relations. Since the various types of semantic knowledge exist in the knowledge structure, the semantic memory should be able to represent different types of knowledge and model their possible interactions and cooperations.

Due to the co-evolution characteristic of episodic and semantic memory learning, this research follows the view presented in [12] that high-level semantic knowledge is learned through generalizing low-level instances learned from episodic memory. Since the episodic memory holds a tremendous amount of past experiences, this research should promote a fast and efficient selection procedure among stored events and episodes in order to form meaningful and different types of semantic concepts and knowledge. The learned semantic memory should serve as a better understanding of environment and lead to more responsive actions and behaviors while guiding the formation of new episodic memory. Resulted from the independent learning of episodic and semantic memory, this research on declarative memory system presents both instance-based information from episodic memory and generalized semantic knowledge. The ultimate question should be answered in this research is that how these different but intrinsically-related memory modules can collaborate for better functioning and performance.

This research should build a procedural memory model, which is able to learn the explicit association between the situation and all possible actions. Upon explicit modeling of each individual component of the long-term memory, this research should further propose a novel cognitive architecture incorporating the learning and processing among the three types of memory. The architecture should show that how the intelligence is usually raised through the complex interactions among the various types of memories. There are two basic paradigms of memory learning in our brain: one is the statistical and incremental learning to develop procedural and semantic knowledge; the other is the instance-based and one-shot learning to store specific past experiences as episodic traces. Therefore, the multiple memory systems should allow mutually incompatible information and knowledge to be learnt and stored in different manners as suggested by the nature of each individual memory modeled, so that they can be used in their individual best fitting situations.

1.4 Approach and Methodology

This research presents a biologically inspired multi-memory framework for modeling the structures and connections between the declarative and procedural memories. The multi-memory architecture is based on fusion Adaptive Resonance Theory (ART) [34] which applies unsupervised learning to categorize input patterns. Fusion ART employs bi-directional processes of categorization and prediction to find the best matching category (resonance). It also learns continuously by updating the weights of neural connections at the end of each search cycle. Fusion ART may also grow dynamically by allocating a new category node if no match can be found. This type of neural network is chosen as the building block of our memory model as it enables continuous formation of memory with adjustable vigilance of categorization to control the growth of the network and the level of generalization. By applying fuzzy operations and *complement coding* [34], fusion ART can also generalize input patterns dynamically and capture a range of values every time it learns.

Using fusion ART as the building block, the architecture includes a highly robust declarative memory system for dynamically encoding and retrieving episodic traces of events and a mechanism for consolidating them into more permanent and general forms in semantic memory. The episodic memory, based on fusion ART network, extracts key events and encodes spatio-temporal relations between events by dynamically creating cognitive nodes in response to a continual stream of sensory input and feedback received from the environment. The model further incorporates a novel memory search procedure, which performs parallel search of stored episodic traces continuously with potentially noisy and imperfect memory cues. The model also includes a forgetting mechanism to remove irrelevant information and prevent the memory from overloading.

This research also proposes an additional knowledge transfer process, wherein the information stored in the episodic memory can be consolidated to produce more general and abstract knowledge in semantic memory. The semantic memory, based on the same self-organizing principle as the episodic memory (i.e. fusion ART), extracts related general facts, meanings and concepts through creating and learning of category codes. Essentially, episodic memory serves as a long-term temporary buffer for rapidly storing events and episodes, which can be recalled at a later time through a memory consolidation process to gradually extract and learn general facts and rules as semantic memory. In this way, the declarative memory supports independent memory running and learning in parallel but at different paces, wherein episodic memory supports rapid, specific and automatic learning while sematic memory provides slow, gradual and incremental learning.

Finally, this research completes the multi-memory architecture with an explicit model of procedural memory. Based on a specialization of three-channel fusion ART, the procedural memory model acquires action-based knowledge and skills through reinforcement learning. In the multi-memory architecture, two major types of memory interaction processes between declarative memory and procedural memory are identified and formalized, wherein the factual information and general knowledge in semantic memory is retrieved to guild the development of various types of procedural skills. As illustrated in Figure 1.2, the architecture consists of four main components, namely, the working memory module to share information and knowledge among all other components, the procedural memory module, the declarative memory module, consisting of episodic memory and semantic memory, and the intentional module to maintain a set of goals in hand and regulate the decision making. Each of the long-term memory modules in our system is built based on fusion ART network [34]. This research further shows how the interaction among various memory systems enable the model to exhibit more versatile decision making and problem solving. Specifically, we identify and formalize two main types of memory interaction and knowledge transfer processes between semantic memory and procedural memory.

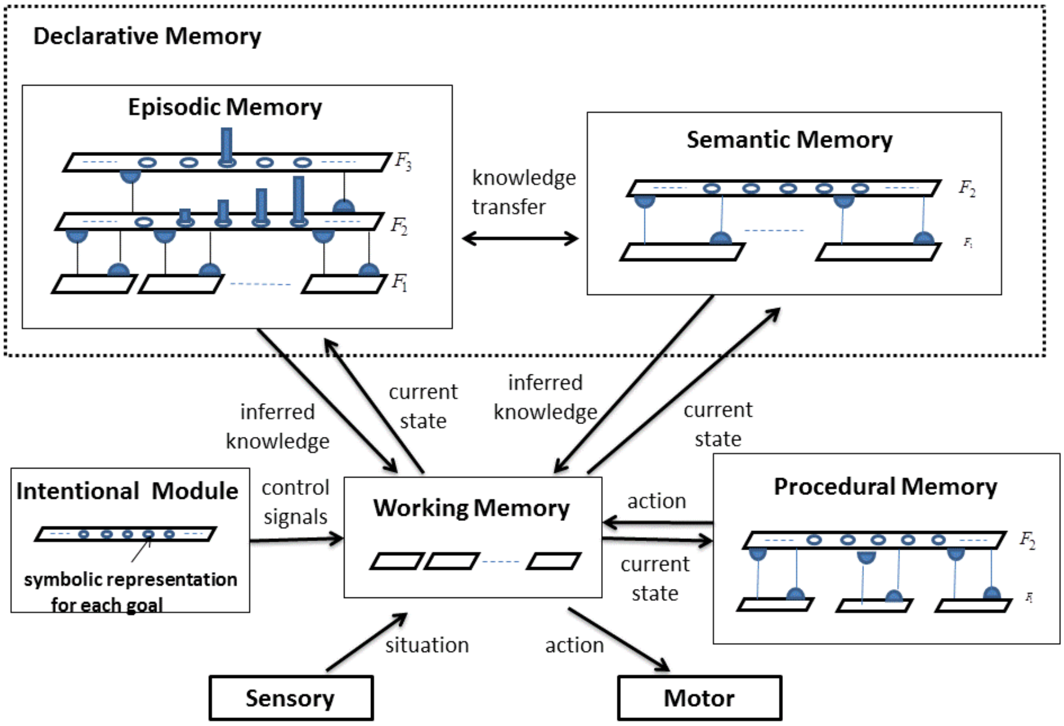


Figure 1.2: The generic framework of the multiple memory systems in a cognitive architecture.

We have conducted empirical experiments in three phases. In the first phase, we investigate the performance of the standalone episodic memory model. We evaluate the robustness of the episodic memory compared with some other memory models in solving a word recognition task [35], as well as a sign language recognition task [36]. The comparisons are also conducted in the Unreal Tournament video game application [37] through various memory retrieval tests. Extended with the forgetting mechanism, the episodic memory is further investigated for its retrieval accuracy and robustness against noisy environments under the application domain of Unreal Tournament.

In the second phase, we evaluate the dual episodic-semantic memory model in the Unreal Tournament domain to support a Non-Player-Character (NPC) agent to learn from experience and improve performance. Through our experiments, we find that our episodic memory model provides a more robust level of performance in learning and retrieving spatio-temporal patterns than other existing types of spatio-temporal memory system. Furthermore, we also find that, as a co-evolving dual memory system, the model does not just improve the task performance, but in some cases, a faster forgetting rate even results in faster learning.

In the third phase of the experimental evaluation, we have investigated the overall performance of the entire multiple memory architecture. We have conducted the experiments on two problem domains: (1) the Toad and Frog puzzle and (2) a strategic game known as Starcraft Broodwar. Our experimental results show that the architecture is able to learn procedural knowledge for the various tasks accross the different game domains based on reinforcement learning signals from the environment. More importantly, the results show that the interaction between declarative memory and procedural memory can lead to a significant improvement in both learning efficiency and performance.

1.5 Organization of Report

The rest of this thesis has been organized as follows. Chapter 2 discusses prior works on modeling declarative and procedural memory, as well as the study on their interaction. Chapter 3 presents the novel episodic memory model with the associated algorithms and procedures developed to learn the individual's episodic experiences. These methods are evaluated through the investigations on the memory performance in various benchmark problems. Chapter 4 proposes a new model of semantic memory, combining with a general procedure of memory consolidation from episodic memory to semantic memory. A performance evaluation on this integrated declarative memory system is also provided in Chapter 4 by applying the system into the Unreal Tournament game environment. Chapter 5 completes a novel cognitive architecture with an additional model of procedural memory based on reinforcement learning. The architecture further integrates with two forms of declarative-procedural memory interactions. The experimental evaluation on the multiple memory systems is based on two problem domains: the Toad and Frog puzzle and a strategic game known as Starcraft Broodwar. Chapter 6 concludes the current research and highlights future work.

Chapter 2

Literature Review

As an essential component of all kinds of learning, the brain's long-term memory has been widely studied in various fields, including artificial intelligence, cognitive psychology and neurobiology. Over the past decades, many models have been proposed using various computational principles and modeling paradigms. This chapter presents a review of these existing computational models on each type of the long-term memory systems, namely episodic memory, semantic memory and procedural memory, as well as the related studies on their interactions. The rest of this chapter is organized as follows. Section 2.1 to Section 2.3 discuss and compare the existing works on modeling episodic, semantic and procedural memory respectively. Section 2.4 and Section 2.5 provide a review of the related works on modeling the memory interaction.

2.1 Models of Episodic Memory

Episodic memory refers to the long-term memory stores one's specific experience in the form of events, as well as their temporal-spatial relations known as episodes [11]. Besides referring to episodic events containing what, where, and when information, episodic memory usually incorporates *autonoetic consciousness* or awareness of the retrieved event as a veridical part of the rememberer's own past existence [38]. This allows one to mentally travel back in time to the past while being aware that the recollection is actually something of the experience of the earlier time [39]. Over past several decades, researchers have identified that episodic memory is critical to support various cognitive capabilities, including goal processing, concept formation and context encoding [13, 14]. Based on the nature of their individual knowledge representation, the existing computational models of episodic memory can be divided into two main categories, namely the symbolic models and the connectionist models.

In a typical symbolic model of episodic memory [40, 19, 41], each episode is encoded as an individual memory trace. The episode retrieval is conducted through a item-by-item similarity search among all the traces stored by explicitly providing a memory cue. Since these models encode an episode as a linearly ordered sequences of individual event traces, they are limited to to explore complex relations between events (e.g. repeated events on an episode), especially in potentially imperfect or noisy environments. Although few symbolic models [42, 19, 43] have employed statistical methods to deal with imperfect and noisy cues, they still consider the memory trace as continuous series of events with no coherent representation of episodic chunks as units of experience. Another approach extends an cognitive architecture, known as the Soar architecture [44, 45], with a novel episodic memory model [1]. With a tree-like storage of episodic traces shown as Figure 2.1, the model makes use of the built-in operations of the Soar architecture to conduct complex memory encoding and retrieval. One critical issue of this approach is that it requires some effective partial matching to deal with incomplete and possibly degraded cues for retrieval [46].

The second family of episodic memory models focuses on understanding the underlying neural structure which forms the basis of episodic learning. Most of these works employ an explicit connectionist modeling of the hippocampal region, which is the brain area commonly thought to be associated with episodic memory. Grossberg and Merrill

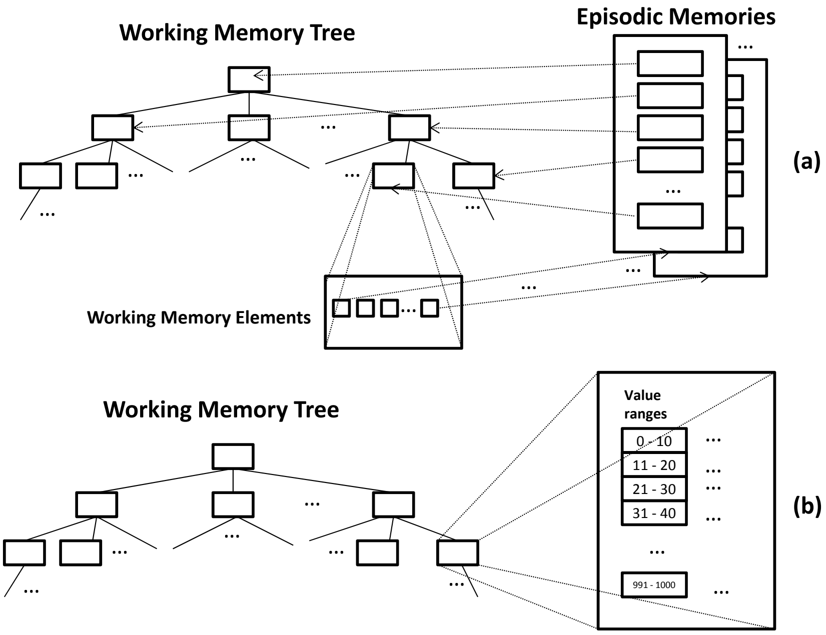


Figure 2.1: Data structures for modeling episodic memory used in (a)instance-based memory matching approach and (b) interval-based memory matching approach in Soar adopted from [1]

combine ART (Adaptive Resonance Theory) neural network with *spectral timing* encoding to model timed learning among different eventual situations in hippocampus [47], regardless other aspects of episodic memory (e.g sequential ordering, multimodal association). A similar model presented in [48] joins an ART network with a Fuzzy Associative Memory (FAM) in a hierarchical manner. The top FAM network learns the temporal relations among events encoded by the bottom ART network. Samsonovich and Ascoli employ a single-layer associative network to study the roles of episodic memory during navigation [20]. In [20], the investigated navigation task is described as the learning of a ordered location sequence from the starting location to the final destination. In this model, each location is represented by a node in the network. It can learn the location sequence by strengthening the node connections between any two adjacent locations within the sequence. A similar model [49, 50] combines two identical ART1 neural networks to share the same category layer (F_2 layer). This category layer learns the association between two spatiotemporally-closed events, each of which is learned by one of the two ART1 networks.

System for the Memorization of Relational Instances from Temporal Impulses (SMRITI) supports complex relational event representation as a group of role-entity binding [2], based on the neural structures illustrated in Figure 2.2. The model can provide a robust event learning in which the memory cues can involve transient values while retrieving with partial information. Although SMRITI has already supported complex inferences on top of the relational representation, it still omits to study the temporal or sequential relations between events. Following the same anatomical outline of hippocampus (including interconnected subarea of entorhinal cortex, dentate gyrus, CA3 and CA1), Norman and O'Reilly [51, 52, 53, 54] develop a biologically detailed episodic memory model [55], namely Complementary Learning Systems (CLS) model, for rapid and automatic episodic memorization. By employing the simple Hebbian learning, CLS model learns episodic memory as a set for highly sparse and separated patterns. The model incorporates a novel learning process of pattern separation to avoid catastrophic interference.

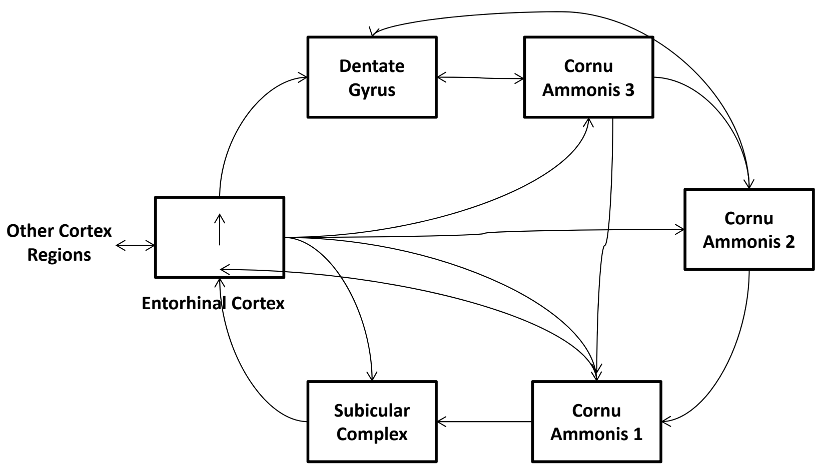


Figure 2.2: Neural pathways interconnecting the main hippocampal components in the SMRITI episodic memory model adopted from [2]

Some more recent connectionist models of episodic memory focus on handling the encoding of spatio-temporal or multimodal patterns without a explicit referencing to hippocampus. Rinkus builds a sparsely distributed neural network model of episodic memory [27], namely Temporal Episodic and Semantic Memory using Combinatorial Representations (TESMECOR), to rapidly stores spatio-temporal patterns in a distributed manner. The model also provides a robust retrieval mechanism which can support complex sequential representation. Compared to other approaches that handle complex sequences, the growth of storage space is non monotonic as sequences are continuously stored in a distributed representation. However, this sparsely distributed model of episodic memory still retains the sequences as a continuous chain of events rather than chunks of episodes. Starzyk et al. [56, 57, 58, 59, 60] also develop a series of neural network models which perform the anticipation-based spatio-temporal learning. The model can store and retrieve complex sequences as units of episodes. Based on a neural model for complex sequential learning and production [61], the model can tolerate errors from cues and strongly supports partial matching. Another episodic memory model, called Temporal Context Model (TCM), associates each memory trace with the corresponding inner mental context [62]. Each of these gradually changing or drifting context represents the currently active thoughts in our brain. The information retrieval of TCM is based on the cue matching between current context state and the contexts associated with the memory traces in storage. Through reinstating the mental context, TCM model explains and illustrates the long-term recency effect of episodic memory, as well as its corresponding process of strategic memory search.

Some existing models of episodic memory have been embedded into autonomous agents to investigate how the proposed episodic learning can contribute to the tasks of problem solving and decision making. Ho et al. [40] demonstrate that the use of episodic memory can increase the survivability of an agent while seeking for resources in a simulated 'ecological' environment. Nuxol [1] integrates episodic memory model into Soar cognitive architecture and shows that, in an artificial environment of tank battles, episodic memory supports a wide range of cognitive functionalities from sensing to abstract reasoning. Samsonovich and Ascoli [20] apply a simplified hippocampus model to a few task domains and show that episodic memory is able to improve the learning for various tasks, including navigation, the tower-of-hanoi problem solving and memory-based inferences. In [63, 64], Zilli and Hasselmo investigate the demands of an episodic memory (modeled as [65]) while learning six different behavior tasks, including spatial alternation, tone-cued alternation, spatial sequence disambiguation, odor sequence disambiguation, non-matching to position, and non-matching to lever. in In [63, 64], these tasks are described as a set of partially-observable Markov decision processes [66, 67, 68]. The empirical results indicate that the episodic memory model can disambiguate order-dependent and probably alternating situations wherein most reactive reinforcement learning algorithms can hardly solve [64]. In [69], the episodic memory model presented in Chapter 3 has been embedded into an autonomous robot in a office domain to study the task of delivering objects to people.

Table 2.1: Comparison between existing episodic memory models

MODEL	MODEL TYPE	KNOWLEDGE REPRESENTATION	LEARNING METHOD	FEATURES
Soar [1]	symbolic	vector; tree structure	stores the snapshots of active working elements as each nodes in a tree structure	effective partial and noisy memory search; integrated into a cognitive architecture
REMII [19]	symbolic; statistical	vector	a Bayesian calculation of the likelihood that the cue matches to a particular stored memory traces	highly robust to imperfect and noisy cues

MODEL	MODEL TYPE	KNOWLEDGE REPRESENTATION	LEARNING METHOD	FEATURES
SMRITI [2]	connectionist; modeling biological details	architecture with several interconnected neural networks	encodes each trace as several role-entity bindings using recruitment learning	encodes complex relations; robust with noisy cues
Cortico-Hippocampal [5]	biological	predictive auto-encoder	encodes stimulus into compressed inner representation with a predicted outcome	clear definition on integrations with cortex
Spatial Navigation [20]	connectionist; modeling biological details	single layered neural network	explains the navigated process as the series of retrievals along the encoded item-context linkage	explicit context encoding
CLS [53]	connectionist; modeling biological details	architecture with several interconnected neural networks	high sparse and separated patterns of cortical activity with basic Hebbian learning	biological detailed model of hippocampus; widely agreed definition on its interaction with cortical areas
TESMECOR [27]	connectionist; encoding spatio-temporal patterns	sparsely distributed neural network	encodes multi-modal and distributed patterns of event, which links each other through their spatio-temporal relations	handling spatio-temporal patterns; robust retrieval

MODEL	MODEL TYPE	KNOWLEDGE REPRESENTATION	LEARNING METHOD	FEATURES
LTM [56, 57, 59]	connectionist; encoding spatio-tsemporal patterns	sparsely distributed neural network	events are represented by the identically-structured neurons in the network and their spatio-temporal relations are studied as their connections	encodes spatio-temporal patterns; noise tolerant; models memory forgetting process
TCM [62]	connectionist; encoding spatio-temporal patterns	vector	associates each memory traces with subject's inner mental context	drifting context update and encoding; explains the process of forgetting and strategic memory search

We summarize our review on the existing episodic memory models in Table 2.1. As shown in Table 2.1, most existing symbolic models are able to support the effective and robust episodic learning of complex conceptual and/or temporal relations from past experiences. However, few of these works are shown to account for various related behavioral and neurophysiological finding, and hence provide a limited insight of the underlying structures and mechanisms which forms the basis of episodic memory. On the other hand, although some connectionist models of episodic memory indicate possible biological details supporting episodic memory functionalities, some significant details are still missing (e.g. how episodic information moves in and out of working memory [70]). Moreover, the computational complexity of these biologically-inspired connectioist models also raise a significant memory capability problem, and further degrade their performance in practical realtime domains.

2.2 Models of Semantic Memory

In contrast to episodic memory, semantic memory stores meanings, concepts, rules, and general facts [11] unrelated to specific experiences. Distributed all over cortical/neocortical areas of the brain, semantic memory can maintain information more permanently than episodic memory. Various types of models have been proposed to study semantic memory over the past decades. One of the earliest model represents semantic knowledge as simple logical propositions as nodes and links in a *semantic network* [71]. The network studies three types of semantic relations (links) between different concepts (node), including "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.

In Copycat, Hofstadter and Mitchell [72] model the semantic memory (the Slipnet) as a network of related concepts, wherein each concept is represented by a node. The link connecting two concepts encodes their relation them, while the corresponding link strength measures the degree of association. The semantic memory in Copycat changes the link strength dynamically: the link strength between two concepts increase as the type of relation represented becomes more relevant to the current situation in hand and vice versa. In this way, the semantic knowledge relevant to the current context tends to be activated and recalled. A similar type of semantic memory network model called Fuzzy Cognitive Maps (FCM) is also proposed to represent causality between concepts [22]. The fuzzy value of the link represents the strength of the casual condition between the connected two concepts . In Soar [23], the semantic memory is represented as declarative chunks to describe properties of a concepts. Soar keeps track on the historical sum of activation level for each declarative chunk store to reflect the general usefulness in the past. Besides representing meanings as symbolic conceptual relations, other approaches apply statistical methods to learn semantic knowledge. Semantic memory models like Hyperspace Analogue to Language (HAL) [24], Latent Semantic Analysis (LSA) [25], and REM-II [19] learns the correlations between concepts by exploring their statistical co-occurences. The statistical models can handle partial or degraded retrieval cues by applying statistical inferences.

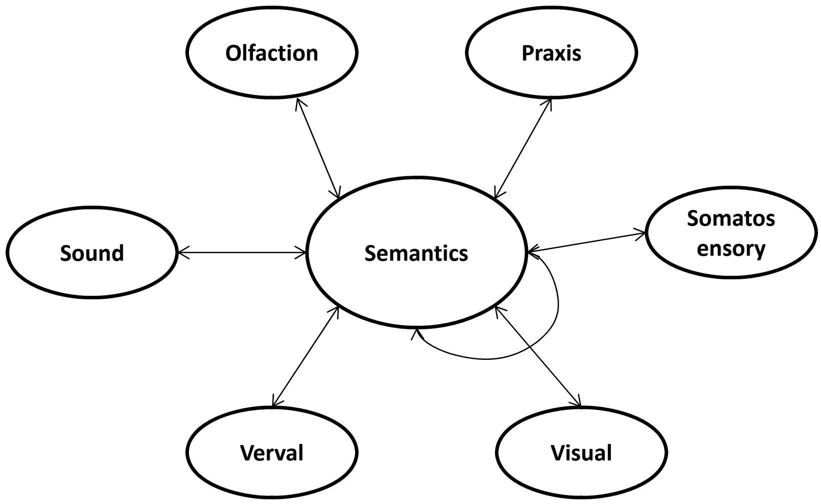


Figure 2.3: Main components of semantic memory and their interconnections in the convergence model adopted from [3]

Besides abstract computational models, some connectionist models are proposed to model the neural structures which forms the basis of semantic memory learning. Hinton [73] proposes one of the earliest connectionist architecture, which emulates the semantic memory by setting up interconnected neural fields to reflect different elements of a proposition. Beyond representing relationships between concepts, the connectionist architecture supports knowledge recollection and generalization through pattern completion accross the network. Further, Rumelhart [28] also develops a similar connectionist model that can automatically learn relational and hierarchical relations among semantic concepts. Using a backpropagation learning method, the model can categorize and discriminate different concepts without external supervision. Shastri and Ajjanagadde proposed the Shruti [74] model, wherein the rational facts (predicate) is represented by the group of dynamic entity-role bindings. The entity-role binding is established through the synchronous firing of the nodes that represent the corresponding conceptual entities and roles. The model supports the process of reasoning and inferencing through the explicitly interconnecting the neuronal patterns between the facts with inferential dependency. Farah and McClelland [26] suggest a bidirectional network model consisting of different interconnected neural fields, wherein each neural field corresponds to one specific type of sensory-functional features. The model is extended further as the convergence theory of semantic memory [3] in which more perceptual and functional features like actions, sounds, and olfactions are incorporated as different neural fields shown by Figure 2.3. Different models and structures above indicate that semantic memory is not a monolithic unitary model but may involve multiple representation and learning mechanisms.

Most semantic models mentioned above are still considered as isolated memory systems that process and acquire semantic knowledge. However, some models also employ episodic memory to be attached with a semantic memory model to form the complete declarative memory systems. Retrieving Effectively from Memory-II (REM-II) [19] connects episodic memory and semantic memory together to learn statistical relationships between items within and accross time. Another episodic memory model based on the Soar cognitive architecture [1] embeds episodic memory directly to the symbolic semantic memory model [75] as additional properties providing contextual and historical information of each assertion and update in the memory. A distributed approach called TESMECOR [27] considers episodic memory as distributed neural connections that also support semantic representation. Although the integrated approaches of episodic memory and semantic memory modeling may provide robust mechanisms to store and retrieve knowledge based on both temporal and relational structures, they still do not reflect the current existing neuropsychological evidences of memory.

The cortico-hippocampal neural model [5] is made to explain why episodic memory and semantic memory are separated in the brain. The model 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 encode the input and the recalled patterns. On the other hand, the neocortex learns and categorizes the input based on the internal representation formed by the hippocampus. In this way, episodic memory and semantic memory can work together to process abstract categorization while they can accurately retrieve specific information. A more realistic model of episodic-semantic memory interaction called Complementary Learning Systems (CLS) [51] reflects the network structure and connections between hippocampus and neocortex in the brain and comprises a particular memory consolidation process. Based on neuroscientific evidences that neurons in hippocampus are reactivated spontaneously during slow wave sleep [76] 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. The model also incorporates a forgetting mechanism of hippocampus in which the strength of neurons are decayed over time before reinstated during or beyond the consolidation process.

Table 2.2: Comparison between existing semantic memory models

MODEL	MODEL TYPE	STRUCTURE	TYPE OF KNOWLEDGE LEARNED	LEARNING METHOD AND FEATURES
semantic network [71]	symbolic	graph describing concepts and their relations	concept categorization; concept hierarchy	simple semantic believes can be learned as stored propositions, connecting two nodes with a certain predicate; supports knowledge generalization and deductive inference

MODEL	MODEL TYPE	STRUCTURE	TYPE OF KNOWLEDGE LEARNED	LEARNING METHOD AND FEATURES
REMII [19]	symbolic	concurrence matrix of features	associative relations among concepts	provides a statistical learning on the co-occurences between memory items; context encoding; context sensitive knowledge acquisition
Rumelhart model [28]	connectionist	three layer neural network with a back-propagation learning	connectionist learning of propositions using in semantic network	encodes the inner representation for each concept; each proposition is trained as a hidden units connecting the concepts and their relations; pattern completion; parallel distributed processing
Farah-McClelland model [26]	connectionist; modeling biological details	bidirectional network consisting of verbal and visual input neural fields	complex and mutltimodal concept	learns each concept as a distributed activation pattern of hidden units sensitively tuned either their functional or perceptual aspects; pattern completion; accounts for behavioral data from the category-specific sematic impairment
convergence model [3]	connectionist; modeling biological details	bidirectional network centered by an intermediating hidden layer connecting each filed of surface neural	complex and mutltimodal concept	the intermediating inner representations encompassing the various concept representation from surface fields; pattern completion; accounts for behavioral data from the semantic dementia impairment

MODEL	MODEL TYPE	STRUCTURE	TYPE OF KNOWLEDGE LEARNED	LEARNING METHOD AND FEATURES
CLS [51]	connectionist; modeling biological details	two-layer network with Conditional Principal Components Analysis (CPCA) Hebbian learning	distributed and overlapping familiarity signals of concept categories	hidden units in the second layer compete and encodes statistical regularities present in input patterns through incremental learning; address the interactions with episodic memory (hippocampus)

We summarize and compare the various types of the existing semantic memory models in Table 2.2. As shown in Table 2.2, the existing works on semantic memory modeling differentiate from each other by their diverse knowledge representations, system architectures, learning paradigms and the hypnosis held on functional organization of semantic memory. Therefore, the current research requires a single unified computational model combining the different aspects of semantic memory , which have been individually addressed, e.g. concept generalization, context representation, temporal encoding and accounting of behavioral data.

2.3 Models of Procedural Memory

Procedural memory refers to the memory of how to perform actions or tasks, for example riding a bicycle or driving a car. Contrasting with declarative memory, most of knowledge in procedural memory are verbally inexpressible, automatically learned and acquired below the level of conscious awareness [30]. The characteristics of procedural knowledge intuitively triggers the thought of representing it as the weighted connections learned through neural networks. The very first connectionist model of procedural memory was developed by Dienes in 1992. In [77], Dienes employs a fully-connected, single layer auto-associative network to encode each executed procedure as a localist pattern, wherein each neuron in the pattern represents the particular position of an action in the temporally ordered action sequence. Due to the simplicity of the network, this model fails to capture complex experiences and support advanced cognition. Cleeremans and McClelland [31] extended the study and stored each action in a sequence as a single pattern. The Simple Recurrent Network (SRN) [4] encodes the temporal context for each action in a sequence by its hidden layer in a backpropagation manner as shown in Figure 2.4. This procedural memory model predicts the next action in an action sequence by presenting all actions within the partial sequence preceding it one by one as network inputs.

Dienes, Altmann and Gao [78] developed a general-propose procedural memory model, wherein the learned knowledge can be transferred among various domains. The model can conduct action prediction on a novel sequence (from a novel domain), as long as the sequence can be explained by the same set of structural knowledge learned from all the trained sequences (from the original domain). The model introduces an addition encoding layer between the hidden layer and input layer of the procedural memory model presented in [4]. Combined the context information learned through the hidden layer, the novel encoding units capture the domain-independent and structural characteristics from a set of training sequences in a particular domain. Then, the learned structural knowledge can be further generalized to other relevant ones. Another model also extends the SRN model by exploring the relations among items temporally nonadjacent (i.e. non-local dependencies, commonly observed in the field of language learning) in [79]. Rather than encoding context information into recurrent layers, the memory buffer model treats several continuous time steps as an input pattern, rather than the single-step input pattern in the SRN model. The model encodes these patterns in a sliding-window manner as the action in the sequence is presented to it one by one. The size of the sliding window is determined by the maximum length of nonadjacent relation (i.e. the maximum time duration considered as related) to be explored.

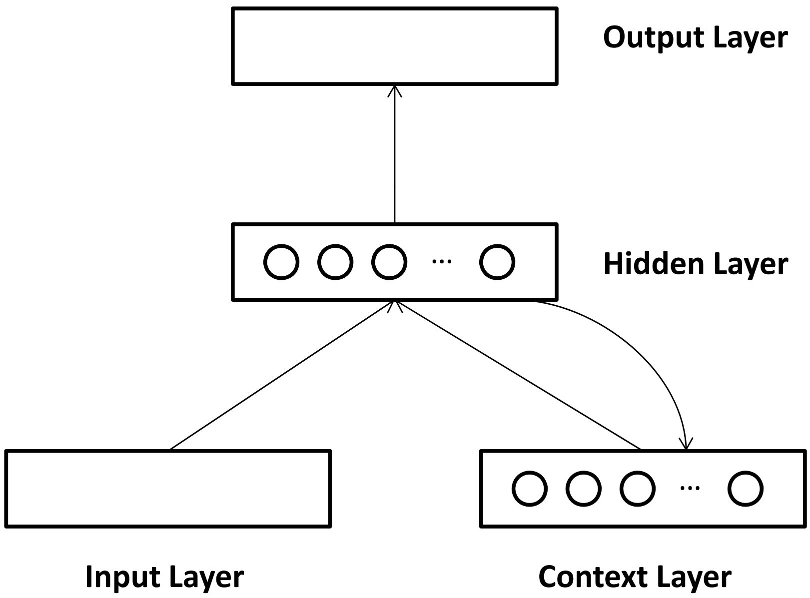


Figure 2.4: The simple recurrent network (SRN) model of procedural memory adopted from [4]

The connentionist models of procedural memory have been applied and investigated in various tasks of artificial language learning, dynamic system control and serial reaction time. The experimental evaluations have confirmed that the performance of a typical connectionist model can roughly fit various characteristics of human's procedural learning in both perfect and noisy test environments. However, these models usually requires some free parameters. Moreover, different configurations on parameter settings can be used to predict totally different set of behavior patterns.

The Temporal Difference-Fusion Architecture for Learning, Cognition, and Navigation (TD-FALCON) [80] models procedural memory to perform reinforming learning, based on an extension of fusion Adaptive Resonance Theory (ART) [34]. Each category nodes in a TD-FALCON network encodes the association among a certain action, a given state and their expected reward. In TD-Falcon, the rewards are leaned by Temporal Difference

(TD) methods to estimate the goodness for a learning system to take a certain action in a given state. They are then used in the action selection mechanism, also known as the *policy*, to select an action with the maximal payoff.

Since the actual procedural learning of human has been observed to always involve certain kinds of memorization on information chunks [81], another family of procedural memory models, namely chunking models, store and manage procedural knowledge in form of memory chunks. In these models, the novel procedural knowledge are learned through comparison and accumulation among all the existing chunks within it. The first chunking model, called Competitive Chunking model, was proposed by Schreiber and Anderson in the context of artificial language learning [82]. In this model, each artificially generated word (or letter string) can be learned by recursively joining its fragments or partial letter string until a single chunk can be used to represent it. During the memory formation, each stored chunk competes with each other to learn the novel word based on its individual memory strength. The memory strength of a memory chunk equals to the sum of its constituent chunks' strength. The strength of each chunk decays along time and is enhanced when it is reused. This model has been further improved [32] by introducing a refined completive learning method to achieve the online attentional control during artificial language learning. Compared to the connectionist models, the chuncking models present significant advantages while resolving the problem of *catastrophic inference*. Consider learning a sequence which shares the starting elements with a perviously learned sequence, the connetionist model usually turns out to wrongly relearn the original sequence through updating the weights. On the other hand, the chunking model can maintain the knowledge on both sequences by allocating additional chucks to represent the novel sequence. However, the chunking models can hardly capture non-local dependencies as illustrated by some connectionist procedural memory models.

Based on a chunk-based representation, both Soar [83] and ACT-R [23] architecture model their procedural memory as a set of *if-then* production rules. Given the current state of memory buffer, sensory inputs and goal settings, both architectures employ their procedural memory to select appropriate actions in order to interact with the environment and accomplish tasks. While ACT-R limits to the selection of a single production rule at any time point, Soar supports the simultaneous and subsequential firing of multiple rules during decision making process. It is to include as much knowledge and information as possible before an action decision is made. To compensate the shortcoming from single-rule firing, ACT-R further models the latency effects [84] of production rule matching, such that the more useful production rules (measured by memory strength) are matched in a shorter time duration and are more likely to be fired and selected. Soar and ACT-R also conduct their procedural memory learning in different manners. In ACT-R, the procedural memory manages its pool of rules through the basic processes of procentralization (i.e. compilation), generalization, composition and analogy. On the other hand, Soar employs the chunking technology [85] to conduct procedural rule learning only in case of knowledge inadequacy. That is, when none of action can be selected during a single decision making cycle (i.e. impasse), Soar resolve the condition by adding and executing additional rule based on its past experiences on similar situations stored in declarative memory modules. The rules learned in Soar can be further refined through its reinforcement learning procedure, which captures and predicts the expected rewards for each state-operator pairs in a continuous manner [86].

More recent research on modeling procedural memory attempts to address a critical issue on how and when these implicit knowledge into can be translated into the corresponding explicit form. This type of research usually holds a point of view that although the procedural learning tries to maintain under the level of consciousness, there is always an attempt during the procedural learning to generalize the learned knowledge into conscious and declarative knowledge. This third category of procedural memory models, namely hybrid models of procedural memory, conducts the procedural learning based on a mixture of both explicit and implicit knowledge representations. Sun et al. proposed a procedural memory model in the Connectionist Learning with Adaptive Rule Induction On-line (CLARION) cognitive architecture, known as the action-centered subsystem (the ACS), to demonstrate how the intelligence can be raised through interactions between the implicit and explicit learning procedures on procedural knowledge [6]. The implicit learning in CLARION is modeled by a three-layer backpropagation neural network, namely Implicit Decision Network (IDN), which is situated in the bottom layer in the ACS [87]. Taking the current state of working memory, sensory buffers and goal structures as inputs, IDN employs a Q-learning algorithm to produce the reactive actions recommendation through the trial-and-error interactions with the environment. The explicit knowledge in CLARION is represented as a set of "condition $\rightarrow$ action" rules, which is learned on top of IDN. Using the action choices in IDN as training samples, the explicit knowledge learning process extracts, refines and deletes (i.e. RER algorithm) its rules in a bottom-up manner. The prior knowledge can also be captured by pre-inserting some fixed rules in the top level. Then the pre-inserted rules can guide the learning in IDN, which also forms a top-down learning paradigm in CLARION. As the implicit and explicit learning procedures mutually independently recommend actions based on their own set of knowledge, the final decision from ACS is a combination of their recommendations through a stochastic action selection method. An action carried out by the ACS is able to change the external states and/or internal states of the architecture. Hence it may trigger the subsequential actions and learning on ACS and other related components.

Table 2.3: Comparison between existing procedural memory models

MODEL	MODEL TYPE	STRUCTURE	KNOWLEDGE REPRESENTATION	LEARNING METHOD AND FEATURES
autoassociator network [77]	connectionist	single layer auto-associator network	localist activation pattern to represent action sequence	encodes action sequence as one localist pattern; each neuron in the pattern represents the occurrence of an action in a particular position of the temporally ordered sequence
SRN network [4]	connectionist	three-layer backpropagation network	one localist pattern to represent each item in the action sequence	encodes context information as the pattern of its hidden layer; action prediction by inputs the context information gathered by all its antecessor
memory buffer model [79]	connectionist	two-layer feed-forward network	localist representation for a action sequence within a sliding window	encodes several action items within several continuous time steps as inputs in a sliding-window manner; explores the non-local dependencies of an action sequence
TD-FALCON [80]	connectionist	fusion ART self-organizing network	commits category node as learning rule specifying the mapping between perception and motor neural fields	perform reinforcement learning through Temporal Difference (TD) methods to estimate the goodness for a learning system to take a certain action in a given state
competitive chunking	chunking [82]	storage of memory chunks	continuously accumulated fragments or chunks	knowledge acquiring by accumulation and comparison among all the previously stored chunks

MODEL	MODEL TYPE	STRUCTURE	KNOWLEDGE REPRESENTATION	LEARNING METHOD AND FEATURES
ACT-R [23]	chunking	set of production rules	if-then production rules	create chunks by rule procentralization, composition, generalization and analogy; update existing chunks based on their general usage; modeling latency of rule matching and its relation to probability of recall; accounts for spacing effects
Soar [83]	chunking	set of production rules	if-then production rules	chunking to learn novel rules and resolve the impasse situations; reinforcement learning on the usefulness of the existing rules
CLARION [6]	hybrid	bottom level: three-layer backpropagation neural networks ; top level: set of rules/chunks	explicit and accessible knowledge as if-then rules; implicit and hidden connectionist representation	bottom level: Q-learning of backpropagation network; top level: rule extraction and refinement; decision made by the combination of output recommendations from both levels

We conclude and highlight our discussion on the existing models of procedural memory by Table 2.3. In general, the existing models represent the procedural knowledge with various abstractness level, based on the different modeling assumptions and being targeted to different applications and tasks. Further research on the procedural memory modeling needs to develop more unified works to put all the pieces together and show more methodological and theoretical insights regarding the procedural learning.

2.4 Modeling Interactions between Episodic and Semantic Memory

As discussed previously, in a declarative memory system, semantic memory is formed based on high level concepts and knowledge, while episodic memory is the collection of low level instances [12]. Although episodic memory and semantic memory represent distinct knowledge and support different cognitive functionalities, they have been commonly recognized to be intrinsically related [17]. Hence, the two forms of memory are interdependent, interacting closely most of the time, each influencing the other in many situations [18]. The interactions between episodic and semantic memory can be generally described using Figure 1.1: while semantic memory can be considered to be the outcome of knowledge transfer from episodic memory, it has been recognized that semantic memory influences our daily activities in understanding as well as interacting with the environment, hence guides the formation of new episodic memory.

From each family of episodic memory model discussed previously, some models have been extended to demonstrate the possible ways of interacting with their corresponding semantic memory models. Basically, the different approaches to modeling interactions mainly reflect the different natures of the corresponding models of episodic and semantic memory. A brief comparison on different episodic-semantic interaction models has been provided in Table 2.4. As an abstract memory model, REM-II [19] connects episodic memory and semantic memory together to learn statistical relationships between items within and accross time. Another abstract episodic memory model integrated the Soar cognitive architecture [1] embeds episodic memory directly to the semantic memory model as additional additional memory storage of the contextual and historical information of each assertion and update in the memory.

The spatio-temporal approach like TESMECOR [27] consider the episodic knowledge as the distributed neural pattern, which also support the knowledge representation of

Table 2.4: Comparison between existing episodic-semantic memory models

Model	Episodic Representation	Semantic Representation	Way(s) of Interactions
REMII [19]	vector of features	concurrence matrix of items	statistical learning of concurrence between different items
Soar	snapshot of active working elements at a certain time	chunks stating concurrence of items	statistical learning of concurrence between different items
TESMECOR [27]	distributed neural patterns on spatio-temporal context	distributed neural patterns on semantic structure	encoded and stored in the same network
cortico-hippocampal [5]	internal representation of input stimuli	category representation of input stimuli	sharing the same inputs; hippocampus supervised the categorization of cortical network
CLS [51]	sparse and separated patterns of experience details	distributed and overlapping familiarity signals of concept categories	consolidation of semantic knowledge initiating by the reactivation in hippocampus
MTT [88]	spatial and temporal context in hippocampus with links to the neocortical regions	context free knowledge in neocortex	consolidation of hippocampal traces during each reactivation

semantic memory. Although the unified approaches of modeling episodic memory and semantic memory may provide robust mechanisms to store and retrieve knowledge based on both temporal and relational structures, they still do not reflect the current existing neuropsychological evidences of memory interations.

The biologically-inspired models of episodic memory attempts to model the episodic-semantic memory interactions through modeling ing the neuronal pathway and functional relations between the hippocampal (episodic) and cortex (semantic) regions. The cortico-hippocampal neural model [5] suggests a simplified interaction procedure as shown in Figure 2.5, such that hippocampus (episodic memory) and neocortex (semantic memory) are two parallel memory systems receiving the same input. The hippocampus learns an internal representation to encode the input and the recalled patterns. On the other hand, the neocortex learns and categorizes the input based on the internal representation formed by the hippocampus. In this way, episodic memory and semantic memory can work together to process abstract categorization and accurately retrieve specific experiences.

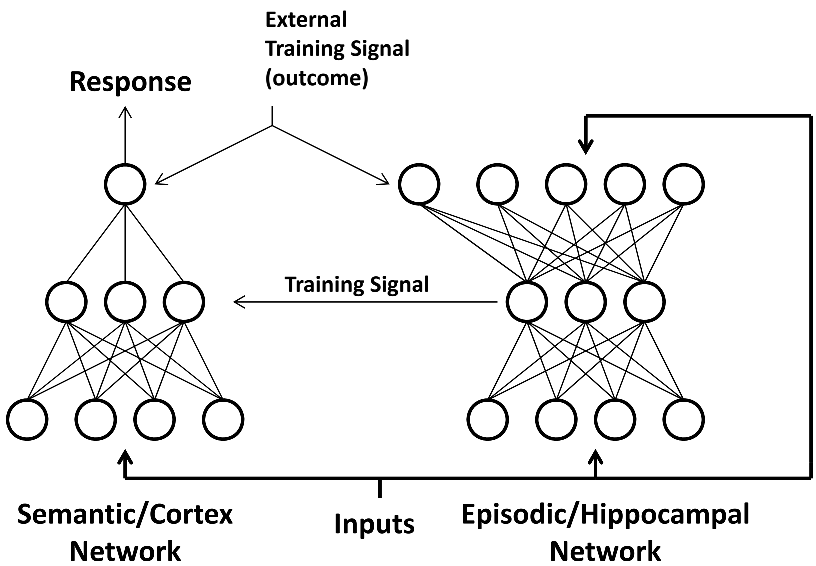


Figure 2.5: The cortico (semantic)-hippocampal (episodic) interaction model adopted from [5]

Some biologically-inspired models explain how the episodic and semantic memory interact with each other by explicitly modeling a process of memory consolidation, wherein each memory trace in semantic and/or episodic memory can be reorganized and stabilized through reactivation after its initial acquisition, in association with each other. Complementary Learning Systems (CLS) [51] proposes a dual memory system which runs both episodic and semantic learning in parallel: hippocampus conducts the fast and automatic episodic learning for the highly sparse and separated episodic memory patterns; while neocortex performs the slow and incremental semantic learning and encodes the more distributed and overlapping semantic patterns. Neuroscientific findings reveal that neurons in hippocampus can be reactivated spontaneously during slow wave sleep [76] and thus reinstates the patterns in neocortex though enacting its slow incremental learning. Based on these evidences for the Standard Model of System Consolidation (SMSC) [89], CLS also emulates an offline consolidation process, wherein the randomly reactivated memory recollection in hippocampus is used as inputs to the neocortex to guide its memory consolidation process. This consolidation process has been further extended to the transformation model [90] by introducing a novel neuronal replay between episodic and semantic memory, wherein either of the dual memories may be dominant depending on the circumstances. Hence, the interaction models based on SMSC state that the formation of semantic memory always requires the episodic memory as its “teacher” and the semantic knowledge is learnt through the explicit transfer process from episodic to semantic memory. Nadel and Muscovites have proposed an alternative model of episodic-semantic memory consolidation, namely Multiple Trace Theory (MTT) [88], which claims for a semantic memory model with less dependance with its episodic component. Nadel and Muscovites argue that the neocortex stores only the context-free (semantic) content, while the complex spatio-temporal (episodic) context is managed in the hippocampus with the links to its corresponding semantic content in neocortex. The model allocates a novel memory trace in hippocampus at each time the episodic memory is (re)activated. Therefore, the retrieval and formation of contextually rich episodic traces in MTT model requires the interaction within the overall hippocampal-neocortical region. However, the remote semantic knowledge can be properly retrieved independent from the hippocampus region after the years of memory consolidation. A more recent work on Competitive Trace Theory (CTT) [91] further extends the ideas from MMT model with a novel decontextualization process during the memory consolidation from episodic events to semantic traces. In the proposed decontextualization process, as 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 events mutually inhibit each other such that none of them can be retrieved and hence "decontextualized" from their semantic representation.

2.5 Modeling Interactions between Procedural and Declarative Memory

As the two major components of the long-term memory systems, procedural and declarative memory involve complex interactions with each other to facilitate our learning and decision making. One generally accepted view on procedural-declarative interaction is that the declarative memory usually initiates and provides foundations for the corresponding procedural learning [92]. For example, while driving is usually considered as the skills developed through procedural learning, it still requires some relevant declarative knowledge, for example "what are the accelerator and brake for?". Therefore, most of cognitive skills developed from our daily activities are resulted from the collaborative learning of both procedural and declarative memory. Besides collaborating with each other, the two memory models also show some level of competition during the process of decision making. That is, procedural and declarative memory compete with each other for activation in order to contribute to the decision making. One possible example on their competition is that, while performing some familiar activity, the inertia brought from our old habits usually blocks the influence from the novel declarative knowledge. The contributions of the procedural or declarative memory in the process of decision making usually depend on the various relevant factors, such as the nature of the task, the subject's familiarity to the environment and the the current emotional states. Moreover, their competition is also observed during the memory formation process. Some behaviorial experiments has identified that there are some resource (neurons) usually shared by the two memories in order to form their individual knowledge representation [93]. Following that, researchers also observed that they block each other during their individual memory consolidation, indicated their reciprocal neuron activities [94].

Various models have been proposed for the development of integrated memory systems, which consists of both procedural and declarative memory. A sample list of such declarative-procedural memory models is given in Table 2.5. Many of these models have been further embedded into autonomous agents for performance evaluation. However, these models typically only study limited forms of declarative-procedural memory interactions, in comparison with those discussed in the previous paragraph.

ACT-R [23] is a cognitive architecture to simulate human cognition, based on the sets of empirical evidences from cognitive psychology and brain imaging. ACT-R assumes human knowledge consist of two separate but mutually related sets of knowledge, procedural and declarative knowledge. As discussed in Section 2.4, declarative memory in ACT-R takes the chunk based representation. Through a competitive learning similar with [82], declarative memory holds the current set of known facts and concepts, as well as active goals. Procedural memory, on the other hand, is modeled by a centered processing system, wherein a set of production rules can be stored and executed. Each production rule is represented by a *"if-then"* statement. Each production rule defines how

Table 2.5: Comparison between existing procedural-declarative memory models

Model	Procedural Knowledge	Declarative Knowledge	Interactions Model	Decision Making
Soar [83]	production rules	semantic chunks: concurrence of features or items overtime ; episode: snapshot of active working elements at a certain time	declarative memory serves as context information to firing production rules; retrieval of declarative memory to form novel procedural knowledge through chunking	five-phase of cycle of input, elaboration , decision , application and output
ACT-R [23]	production rules	declarative chunks	declarative memory serves as context information to firing production rules	action recommendation by firing a single production rule
CLARION [6]	top level: explicit and accessible knowledge as if-then rules; bottom level: implicit and hidden connectionist representation	top level: declarative chunks encoded by general knowledge store (GKS) and associative rules; bottom level: auto- and hetero-associative connectionist representation	declarative memory is under the control of procedural memory; meta-cognition take the overall supervision of their interactions	action recommendation through combining the action decision making from both the top and bottom level of action-centered subsystem, under the supervision from the motivational and meta-cognitive subsystems

a certain goal can be accomplished if its associated pre-conditions can be satisfied, that is the pre-conditions can be matched with the currently activated contents from declarative knowledge. Therefore, the basic reasoning process of ACT-R can be simply described as the following steps: (1) at any point of time, an ACT-R based agent attempts to accomplish a task by activating a particular goal representation in its declarative memory; (2) based on the current set of activated declarative knowledge, the agent selects and fires a single production rule from its procedural memory; (3) the firing of a production rule can lead to either directly accomplishing the goal, or the subsequential activation and/or learning of more declarative chunks which may further trigger the firing of additional production rules; (4) step (2) and (3) are repeated until the goal is accomplished. The recursive process of rule firings illustrates the step-by-step rationale for problem solving commonly observed in our daily life. Hence, ACT-R demonstrates how the intelligence has be raised during the interactions between declarative and procedural memories.

Similar to ACT-R, Soar [83] models the process of problem solving by managing on a complex set of goals, states and actions. The memory system of Soar includes both the long-term memory, consisting of the procedural, semantic and episodic memory, and the working memory. The working memory model in Soar holds all the available knowledge and information about the current situation, including the goals, the perceptions on the external environment and the lists of available actions. Driven by the goal of solving the current problem in hand, Soar selects actions to take, namely *operator*, from a set of *"if-then"* production rules, which represents the pool of its procedural knowledge. Contrasting with the decision making in ACT-R, Soar allows the unlimited number of productions rules to fire in parallel, that is to take all the relevant procedural knowledge into consideration before make a decision. After all the related rules have been fired, Soar will select a single operator to take according to its specific preference evaluation of rules.

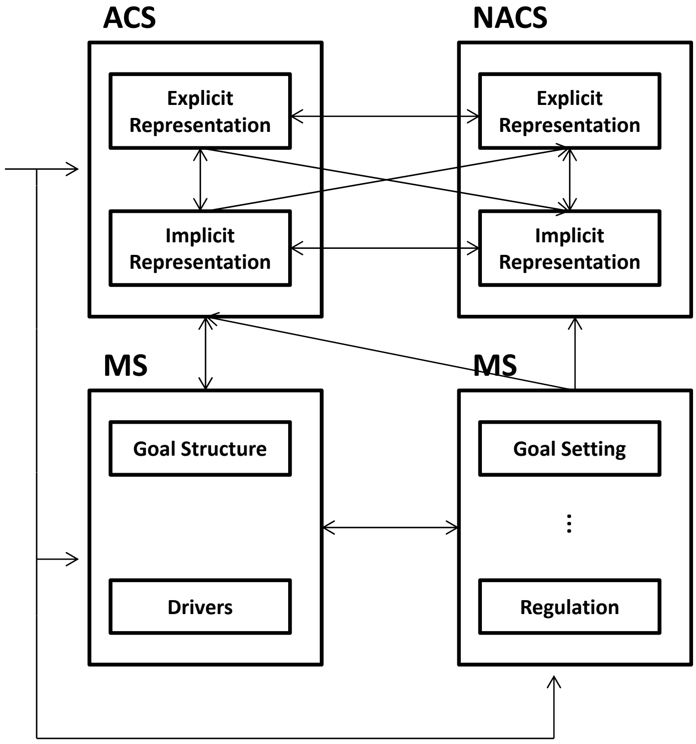


Figure 2.6: The CLARION model adopted from [6, 7]. ACS denotes the action-centered subsystem, NACS the non-action-centered subsystem, MCS the meta-cognitive subsystem, MS the motivational subsystem

From the decision making process described, the behaviors of a Soar-based agent is mainly guided by its procedural memory. However, the semantic and episodic memory may play the critical roles when the current level of procedural knowledge is insufficient. This is signaled by multiple or no operators recommended during the process of the rule firing. In this case, the Sore-based agent will continue its decision making by looking into its similar experiences from the semantic and episodic memory to resolve the impasse. Once sufficient knowledge has been retrieved to choose the correct operator, a new production rule will be directly encoded into its procedural memory in a manner of *chunking*. In general, Soar supports the parallel learning and execution for procedural and declarative memory: while procedural memory is automatically retrieved during each cycle of decision making and executes the highly skilled behaviors; the learning of semantic and episodic memory is continually conducted without the interference from the decision making, and the access of the declarative memory is only through explicit memory retrievals in the scenarios of "non experts".

As shown in Figure 2.6, CLARION [6] cognitive architecture organizes its procedural and declarative memory into two separated subsystems, the action-centered subsystem (ACS) [87] and the non-action-centered subsystem (NACS) [7] respectively. In CLARION, declarative memory is under the complete control of procedural memory, illustrated by the following two aspects: (1) declarative memory provides necessary reasoning outcome needed for the decision making process conducted by procedural memory and (2) the context and declarative information about the current situation is also retrieved by declarative to facilitate the decision making process. Moreover, CLARION suggests an additional supervisory subsystem, namely meta-cognitive subsystem (MCS) [95], to monitor and control the interactions among the two long-term memory systems.

Chapter 3

Modeling of Episodic Memory

In this chapter, we present an episodic memory model supporting the encoding of an individual's experience as well as their spatio-temporal relation, based on a generalization of fusion Adaptive Resonance Theory (ART) [34], as published in [96, 97]. The model called EM-ART supports complex-event storage through its multiple-channel pattern learning capability inherited from fusion ART. A special encoding scheme is also introduced that allows complex sequences of events to be grouped and recognized. EM-ART further incorporates a novel memory search procedure, which performs parallel search of stored episodic traces continuously in response to potentially imperfect search cues. In addition, the model employs a mechanism of gradual forgetting so as to maintain a manageable level of memory consumption over a possibly infinite time period crucially needed by a real-time system.

We have conducted experimental studies on the model through two different applications. The first application is a word recognition task, wherein the 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, our experiment results show that the model produces an equal or better retrieval performance. We also evaluate the model in a first person shooting game called Unreal Tournament. In the game, the model is used to learn episodic memory based on an agent's encounters. Experiments show that the model provides a more robust level of performance in encoding and recalling events and episodes using various types of input queries involving incomplete and noisy cues, compared with another spatio-temporal memory model. By further examining the effects of forgetting, we find that the incorporated forgetting mechanism also promotes a more efficient and robust learning by continuously pruning of erroneous and outdated patterns.

The rest of this chapter is organized as follows. Section 3.1 provides an introduction to the modeling of episodic memory. Section 3.2 presents the architecture of our episodic memory model. Section 3.3 and section 3.4 present the algorithms and processes for event and episode encoding and retrieval respectively. Section 3.5 discusses the forgetting mechanism incorporated in the model. Section 3.6 and section 3.7 investigate the performance and robustness of the model in the word recognition task, the sign language recognition task and the shooting game respectively. Section 3.8 provides a brief discussion and comparison of selected work on episodic memory models. The final section concludes and highlights future work.

3.1 Introduction

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 that one experiences.

3.1.1 Memory Formation

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 characteristics 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.

3.1.2 Memory Retrieval

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

- **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.

- **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 present event sequence has missing parts in the event representation. Two basic requirements of episode recognition include: (1) tolerance to incomplete cues, which only form part of the stored episodes and (2) tolerance to errors in situational information, for example, noise in event attributes and variations in the order of event sequences.

- **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.

3.1.3 Forgetting

Many studies (e.g. [98, 99]) have indicated that the memory traces in hippocampus are not permanent and occasionally transferred to neocortical areas in the brain through a consolidation processes. It 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. The forgetting in episodic memory also brings strong evidence to the knowledge transfer process from hippocampus to neocortical memory. The memory consolidation process makes the memory traces less prone to disruption and forgetting [2].

Forgetting is not only a natural and desired characteristics of biological intelligence. It is also a prevalent operation in continuous operation of real time artificial models that gradually learns 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 memory operations of episodic memory in real-time environments. However, as episodic memory is event driven, the memory stores events that could be separated in unpredictable intervals along time. Therefore, it is unrealistic to incorporate a forgetting process for episodic memory relying on time as the only or the most important regulator of its functions.

In this research, we describe the basic principles of forgetting as follows: in the formation of episodic memory, short term memory events and/or episodes that were not rehearsed frequently and did not have significance will be quickly forgotten, while those that are significant or repeated frequently will last longer. In order to achieve these requirements, the dynamic memory architecture may include the following two main components: (1) an evaluation method on the importance of stored events and episodes based on various related criterions (such as rewards and accessing frequency); (2) a forgetting mechanism continuously enhances the storage of important past experiences while purging the aging and trivial situations.

3.1.4 Summary

Taking the above into consideration, an episodic memory model should satisfy the following basic requirements:

- Efficient event representation, which is able to describe complex situations and events;
- Efficient episode representation, which explores spatio-temporal relations among events which form the episode;
- Well-defined generalizations on representations, which accurately distinguishes critical and irrelevant differences among them (for both events and episodes);
- High error tolerance to incomplete or noisy cues;
- Fast memory operations, including memory encoding and retrieving;
- Tracking the importance of events and episodes in realtime based on rewards, surprises, emotions, interpretation and access frequency; and
- Forgetting mechanism to deal with the limited capacity issue.

3.2 The Model

Our episodic memory model, called EM-ART, is built by hierarchically joining two multi-channel self-organizing neural networks, called fusion ART networks. Based on Adaptive Resonance Theory (ART) [100], fusion ART dynamics offers a set of universal computational processes for encoding, recognition, and reproduction of patterns.

As shown in Figure 3.1, the model consists of three layers of memory fields: F_1, F_2 and F_3. The F_1 layer, connected with the working memory, holds 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.

3.3 Event Encoding and Retrieval

An event consists of attributes characterizing what (e.g. subject, relation, action, object), where (e.g. location, country, place), and when (e.g. date, time, day, night) an event

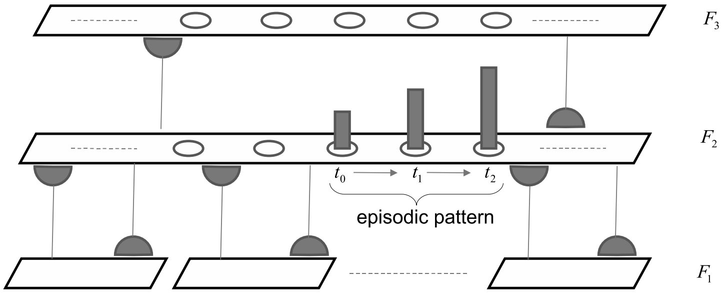


Figure 3.1: The three-layer neural network architecture of the episodic model: F_1 is the input layer connected to the working memory, F_2 for event recognition, F_3 for episode recognition

occurs. Figure 3.2 shows an example of the structure of an input event based on the Unreal Tournament domain [37]. This structure is also used in the experiments for evaluating EM-ART (explained in later sections). In the structure shown, the location is expressed using a 3-dimensional 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. killing and being damaged) is presented to the model as a reward value. Information about time is not included in this case, but it can be assumed that the temporal information has been represented inherently in the episode.

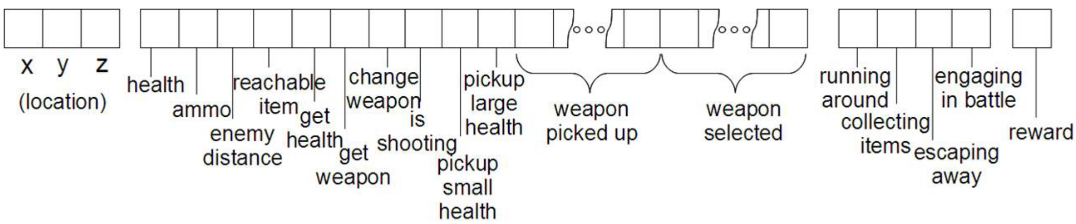


Figure 3.2: The vector representation of an event based on the Unreal Tournament domain

3.3.1 Fusion ART

Fusion ART network [34] 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 multi-channel input vector. Figure 5.2 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.

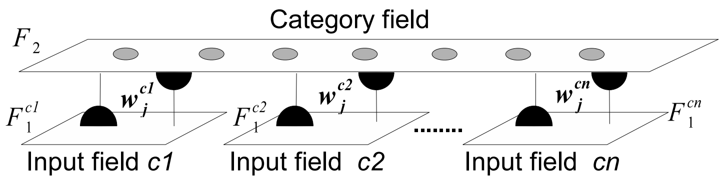


Figure 3.3: The fusion ART Architecture

For completeness, a summary of the fusion ART dynamics [34] 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 and $i > 1$ 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.

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

$$T_j = \sum_{k=1}^{n} \gamma^k \frac{|\mathbf{x}^k \wedge \mathbf{w}_j^k|}{\alpha^k + |\mathbf{w}_j^k|}, \tag{3.1}$$

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

$$T_J = \max\{T_j : \text{for all } F_2 \text{ node } j\}. \tag{3.2}$$

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

$$m_J^k = \frac{|\mathbf{x}^k \wedge \mathbf{w}_J^k|}{|\mathbf{x}^k|} \geq \rho^k. \tag{3.3}$$

If any of the vigilance constraints are 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:

$$\mathbf{w}_J^{k(\text{new})} = (1 - \beta^k)\mathbf{w}_J^{k(\text{old})} + \beta^k(\mathbf{x}^k \wedge \mathbf{w}_J^{k(\text{old})}). \tag{3.4}$$

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.

3.3.2 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 Figure 3.2. Using the standard operations of fusion ART, the recognition task can be realized by a bottom-up activation given the input vector. On the other hand, the top-down activation (readout operation) achieves the recall task. Figure 3.4 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 3.1.

Algorithm 3.1 for event recognition and encoding is designed to handle complex sequences involving repetition of events. The iteration condition in line 3 Algorithm 3.1 ensures that the same node will not be selected if it has been selected previously as a

Algorithm 3.1 Event Encoding

```
for each incoming pattern of event in F1 do
    Activate and select a node (through winner-take-all) in F2
    while the node is not in resonant condition or the node has been selected previously do
        Reset the current node activation
        Choose another node in F2
    end while
    if no matching node can be found in F2 then
        Recruit an uncommitted node in F2
        Learn it as a novel event
    end if
end for
```

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 3.3.1. 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.

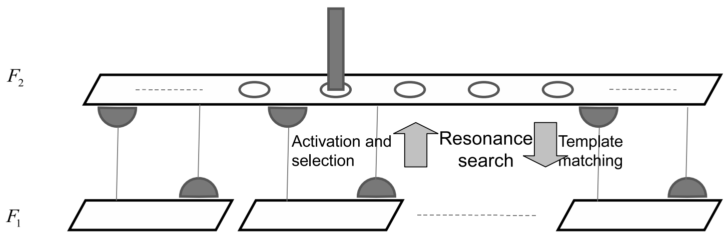


Figure 3.4: Operations between F_1 and F_2 in EM-ART: bottom-up activation to recognize and select an event, top-down activation to recall an event

3.4 Episode Learning and Retrieval

3.4.1 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 Episodic Memory (EM) model in this chapter extends the fusion ART model so that it can associate and group patterns across time.

Specifically, we adopt the method of *invariance principle* [101, 102], 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.

The method has accurately emulated the characteristic of serial learning conforming the psychological data about human working memory [101]. The approach has also been simplified as *gradient encoding* by replacing the multiplication with the adding/subtracting operation and is successfully applied to the intentional Fusion Architecture for Learning, COgnition and Navigation (iFALCON), a belief-desire-intention (BDI) agent architecture composed of fusion ART [103].

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 $t_0, t_1, t_2, ..., t_n$ 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 $y_{t_i} > y_{t_{i-1}} > y_{t_{i-2}} > ... > y_{t_{i-n}}$ 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 STM (Short-Term Memory), in which a recently presented item has a higher chance to be recalled from the memory.

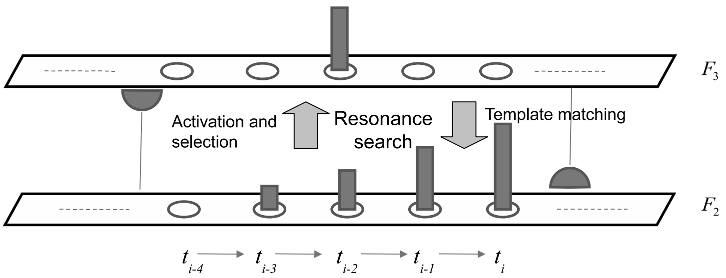


Figure 3.5: Operations between F_2 and F_3 in EM-ART: bottom-up activation to recognize and select an episode, top-down activation to recall an episode

The process of episode learning in EM-ART is shown in Figure 3.5. 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 $y_j^{(new)} = y_j^{(old)}(1-\tau)$, 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 3.2.

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 3.3.1.

Algorithm 3.2 Episode Activation and Learning

```
for each event in an episode S do
    Select a node in F_2 based on the input pattern in F_1
    Set the activation y_j of the selected node to maximum
    Decay activations of all previously selected nodes i by y_i^(new) = y_i^(old)(1 − τ)
    if the event is the end of S then
        Activate, select, and learn a node in F_3 based on the pattern formed in F_2 by
        resonance search
    end if
end for
```

3.4.2 Episode Retrieval

After episodes are learned, a particular episode can be recalled using different types of cues. A cue for the retrieval can be a partial sequence of the 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.

To retrieve an episode based on a weak 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 3.3.

Algorithm 3.3 Episode Recognition

for each incoming event **do**
  Select a node in F_2 based on the incoming event in F_1 by resonance search
  Set the selected node activation y_j to maximum
  Decay the value of every previously selected node i by $y_i^{(new)} = y_i^{(old)}(1 - \tau)$
  Activate and select a node (winner-take-all) in F_3 based on the current pattern formed in F_2
  if the selected node matches with the pattern **then**
    Episode is recognized and the search finishes
    Continue to the next stage of retrieval by exiting the loop
  end if
end for

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. 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. That is, the episode playback starts with the first event in the episode and ends with the last event occurred. However, due the *recency effect* during episode encoding as mentioned in Section 3.4.1, the first event in the episode is encoded with the smallest value in the sequence pattern, while the maximum value is given to the termination event. The EM-ART resolves this issue by vector complementing the values in F_2 before reading out the complete events in F_1. After the sequential pattern is readout 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.

3.5 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 realtime 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 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 a 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:

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

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.

3.6 Benchmark Comparison

3.6.1 Empirical Evaluation on A Word Recognition Task

In this section, we compare the performance of the model with other sequential memory methods to be discussed in Section 3.1 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 [35]: *"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 by one. 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 a 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 train the EM model using all corresponding corrected words indicated by the typoglycemia test. We set the choice parameter $\alpha = 0.1$, contribution parameter $\gamma = 1$ and learning rate $\beta = 1$ for event learning, $\alpha = 0.1$, contribution parameter $\gamma = 1$, learning rate $\beta = 1$ (uncommitted node) and learning rate $\beta = 0.3$ (committed node) for sequence learning. 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 misspelt 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 many other methods including hidden Markov model (HMM), Levenshtein distance method and a spatio-temporal network model called LTM model, (i.e. long-term memory), as reported in [57]. As shown in Figure 3.6, HMM can correctly retrieve 94.67% of the learned words from all words in the test, while the accuracy for Levenshtein distance method is 89.36%. EM-ART and LTM model provide an equal retrieval performance of 100% accuracy for this retrieval test. These results shows EM-ART provides better recognition performance compared to HMM and Levenshtein distance method in the typoglycemia test; LTM has a simlar performance as EM-ART by tolerating all errors while recalling the whole misspelt paragraph.

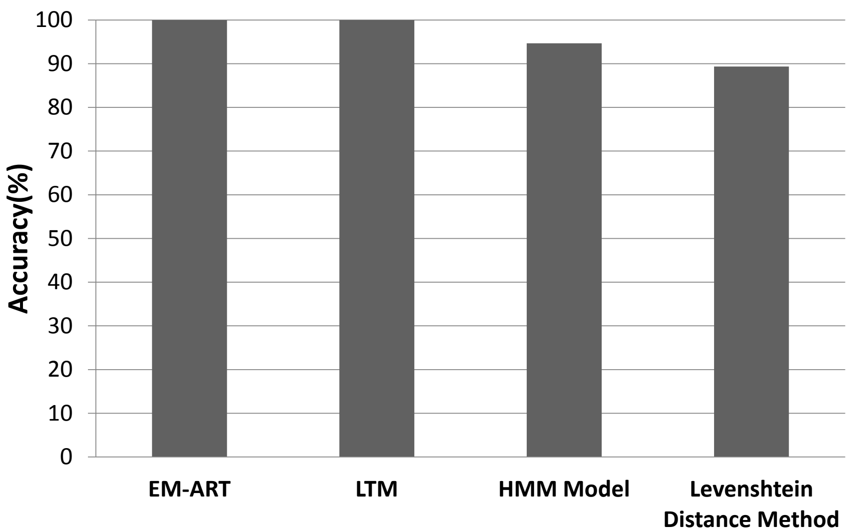


Figure 3.6: Comparison of retrieval accuracy (in %) on the typoglycemia word recognition benchmark

3.6.2 Empirical Evaluation on A Sign Language Recognition Task

In the previous section, the EM-ART model shows a similar performance in the word recognition task compared with the LTM model [57]. In this section, we evaluate the performance of the EM-ART model in a sign language recognition task. The experiment results are compared with the LTM model, as well as one of its extended models presented in [59]. In this task, the EM-ART model learns the sample set of the Auslan (Australian Sign Language) signs presented in [36]. This Australian Sign Language (ASL) data set contain 27 samples for each of 95 Auslan signs. Each sample in the ASL dataset was collected from a native signer with high-quality position trackers and instrumented groves. The sample in the dataset has an average length of 57 frames, each of which contains 22 channels of information (e.g. x, y, z, roll, pitch, yaw and individual figure motions of both hand).

For a proper comparison, we employed the similar set of experiment configuration as stated in [104, 59]. We extracted the trajectory from each sign sample by the first-derivation of the data on the x and y coordinates of both hands. The trajectory is then smoothed by a moving average window of size three. To perform the ASL recognition task, each data point in the smoothed trajectory is fed into the EM-ART model as an input pattern (i.e. an input pattern consist of one input field with four attributes) one by one. As the input patterns are presented one by one along the trajectory, the EM-ART model learns each data point as an event node in F_2 layer. The model eventually learns the entire trajectory of a sign sample as an episode in F_3 layer. The episode describes the temporal ordering of the individual data points along the corresponding trajectory.

We conducted four sets of experiments with the different numbers of signs to learn, i.e. 8, 16, 29 and 38. In each set of the experiments, the EM-ART model is trained using the half of the sample trajectories (13 trajectories per sign) and the retrieval test is conducted by the full set of the samples for the selected sign. During each run of experiments, the signs to be learned is randomly selected. We set the same parameters of the model as stated in Section 3.6.2, except that the vigilance at the event level is set to 0.97 to consider the noisy trajectory inputs (i.e. $\rho^e = 0.97$). We compared the performance with several other methods including HMM [104], Gaussian mixtures model (GMM) [104], self-organizing map (SOM) [105], the original LTM model [57] and the extended model of LTM [59]. The performance measurement of the compared models is reported ias [104, 59]. The experiment results are shown in Table 3.1. From Table 3.1, EM-ART is able to retrieval more testing trajectories accurately compared with HMM, GMM, SOM and the original LTM model, especially as the number of the signs learned increases. Compared with the extended LTM model, EM-ART provides a better retrieval performance in term of the average retrieval accuracies, but with larger values of standard deviations.

Table 3.1: Retrieval accuracies of EM-ART and comparisons with other models in the ASL language recognition task.

Model	No. of classes (no. of runs/no. of test instances)			
	8 (50/216)	16 (30/432)	26 (10/738)	38 (10/1026)
EM-ART	0.9615 (± 0.0487)	0.9170 (± 0.0405)	0.8986 (± 0.0404)	0.8869 (± 0.0390)
LTM [59]	0.9412 (± 0.0244)	0.9009 (± 0.0296)	0.8884 (± 0.0189)	0.8671 (± 0.0147)
Original LTM model [57]	0.8102 (± 0.0409)	0.7676 (± 0.0200)	0.7372 (± 0.0135)	0.7263 (± 0.0346)
HMM [104]	0.86	0.78	0.69	0.66
GMM [104]	0.85	0.74	0.67	0.64
SOM [105]	0.82	0.76	N.A.	N.A.

3.7 Case Study in A Game Domain

3.7.1 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 non-player 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.

In our EM-ART model, an event can be defined as a vector, as shown in Figure 3.2. 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 7 to over 250. The consistent set of parameter values is applied to the experiments in the rest of the chapter if not explicitly stated: the choice parameter $\alpha = 0.1$, contribution parameter $(\gamma^{location}, \gamma^{status}, \gamma^{behavior}, \gamma^{consequence}) = (0.1, 0.3, 0.3, 0.3)$ and learning rate $\beta = 1$ (uncommitted node) and learning rate $\beta = 0.3$ (committed node) for event learning; the choice parameter $\alpha = 0.1$, contribution parameter $\gamma = 1$, learning rate $\beta = 1$ (uncommitted node) and learning rate $\beta = 0.3$ (committed node) for sequence learning. The $\gamma^{location}$, γ^{status}, $\gamma^{behavior}$ and $\gamma^{consequence}$ represent the contribution parameters for the four channels of location, status, behavior and consequence in the event representation as described in Section 3.3 and Figure 3.2.

3.7.2 Episode Retrieval by A Game Agent

We build several exemplar EM models using various vigilance values to access their effect on both episode learning and retrieval. Table 3.2 shows the memory sizes of the EM models based on different vigilance setting, described by the total number of events and episodes in the built models. As reported in Table 3.2, 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.

Table 3.2: Comparisons of the EM model sizes (in numbers of event and episode nodes) at various levels of vigilances.

(ρ^e, ρ^s)	Number of Episodes	Number of Events
$(1.0, 1.0)$	100	6705
$(1.0, 0.9)$	100	6705
$(0.95, 1.0)$	98	2692
$(0.95, 0.9)$	98	2692

After the EM 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; (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 chapter we only present the model performance under a narrower range of vigilance values from 0.9 to 1.0. The results of these tests are reported in the following paragraphs.

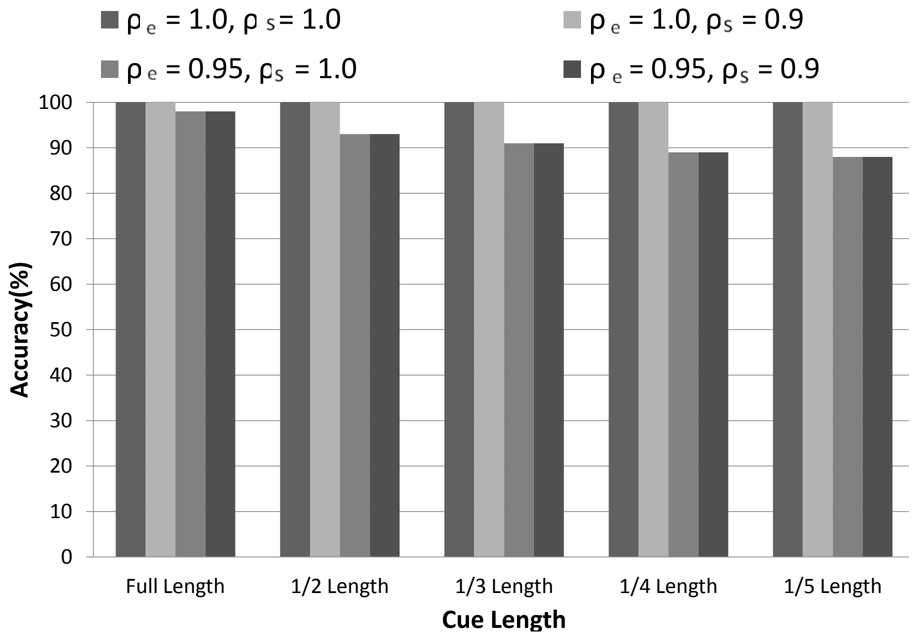


Figure 3.7: Comparison of retrieval accuracies (in %) using partial cues from the beginning of episodes with various lengths under different vigilance values.

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. Figure 3.7 gives the retrieval accuracy using cues of various length under different vigilance values. As shown in Figure 3.7, 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 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 insignificant impacts on the performance due to the data characteristics discussed previously.

Table 3.3: Comparison of retrieval accuracies (in %) using partial cues from (a)the beginning (b) the end (c) arbitrary locations of episodes with various lengths under different vigilance values.

Cue Type	(ρ^e, ρ^s)	Cue Length				
		full	1/2	1/3	1/4	1/5
(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	1/2	1/3	1/4	1/5
(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	1/2	1/3	1/4	1/5
(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

Retrieving from End of Episodes In this retrieval test, cues are extracted from the end 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. The section (b) of Table 3.3 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) comparing with retrieving from the beginning of episodes. The difference in performance can be observed by introducing the multiplicative decay process described in Section 3.4.1. 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.

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. The section (c) of Table 3.3 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.

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 3.4, with a specified error rate.

Table 3.4: Comparison of retrieval accuracies (in %) using full-length cues with various levels of noises on (a) event (b) episode representation.

Cue Type	(ρ^e, ρ^s)	Error Rate				
		2%	5%	10%	15%	
(a) full length cues with various level of noises on event representation	$(1.0, 1.0)$	98	97	91	72	
	$(1.0, 0.9)$	97	97	93	75	
	$(0.95, 1.0)$	95	95	82	57	
	$(0.95, 0.9)$	95	95	85	57	
			5%	10%	15%	20%
(b) full length cues with various level of noises on sequence representation	$(1.0, 1.0)$		100	100	100	100
	$(1.0, 0.9)$		100	100	100	100
	$(0.95, 1.0)$		98	98	98	97
	$(0.95, 0.9)$		98	98	97	97

We test the model with various error rates on event representation and the results are shown in the section (a) of Table 3.4. The test shows that the built model can correctly

Algorithm 3.4 Generation of Noisy Events

Input: Error rate $r \in (0, 100)$

Randomly select $r\%$ events in the original data set
for each selected event in the original dataset **do**
  for each attributes A, in the selected event **do**
    Randomly draw a value e from a Gaussian distribution $N(0, \sigma^2)$
    $A' = A + e$
    if A is boolean value **then**
      if $A' \geq 0.5$ **then**
        $A = true$
      else
        $A = false$
      end if
    end if
    if A is real value **then**
      $A = A'$
      if $A' > 1$ **then**
        $A = 1$
      end if
      if $A' < 0$ **then**
        $A = 0$
      end if
    end if
  end for
end for

retrieve at least 90% of all episodes with an error rate as high as 10%, at an event vigilance of 1.0. However, the performance drops to roughly 70% as the error rate reaches 15%. 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, higher vigilance (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 3.1.

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 3.5, given the desired rate of noise.

We test the model with various error rates on sequence representations and the results are shown in the section (b) of Table 3.4. Similar to the previous results, to achieve tolerance to high level of noise, the model requires a relatively high event vigilance (ρ^e). With a vigilance of 1 at event level, the model can achieve 100% retrieval accuracy with an error rate as high as 20%.

3.7.3 Comparison with A Long-term Memory Model

Since EM-ART and the LTM (i.e. long-term memory) model [57] show similar performance on the word recognition tests described in Section 3.6, in this section, we conduct further performance comparison between these two models by repeating the retrieval tests conducted in Section 3.7.2. Since the retrieval tests show that the performance of

Algorithm 3.5 Generation of Noisy Episodes

```
for each episode S_1 stored do
    Randomly select another stored episode S_2 whose length is longer than S_1
    Randomly set the value of x where 0 < x < the length of S_1
    Set y = x + n, where n/S_1.length is the desired error rate
    Replace partial event sequence in S_1 indexed by [x, y] with the corresponding partial event sequence of S_2
end for
```

both models are almost the same with partial cues and only the retrieval results with noisy cues are presented in this chapter.

Figure 3.8 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. The performance difference between the two models widens when dealing with noises on the event ordering. As shown in Figure 3.9, there is a gradual degradation of accuracy level in the LTM model with the increase in the error introduced. In contrast, EM-ART always retrieves all the correct episodes even though the error rate has reached 20%.

The results above confirm that the 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.

3.7.4 Analysis on Effects of Forgetting

Signals from environment are subject to noises. The episodic memory model should maintain a robust performance while learning with noisy information. Through its one-shot learning, EM-ART encodes all the incoming events into its storage, regardless the validity of the information. To deal with this problem, the incorporated forgetting mechanism discriminatively manages these learned experiences as follows: the noisy experiences

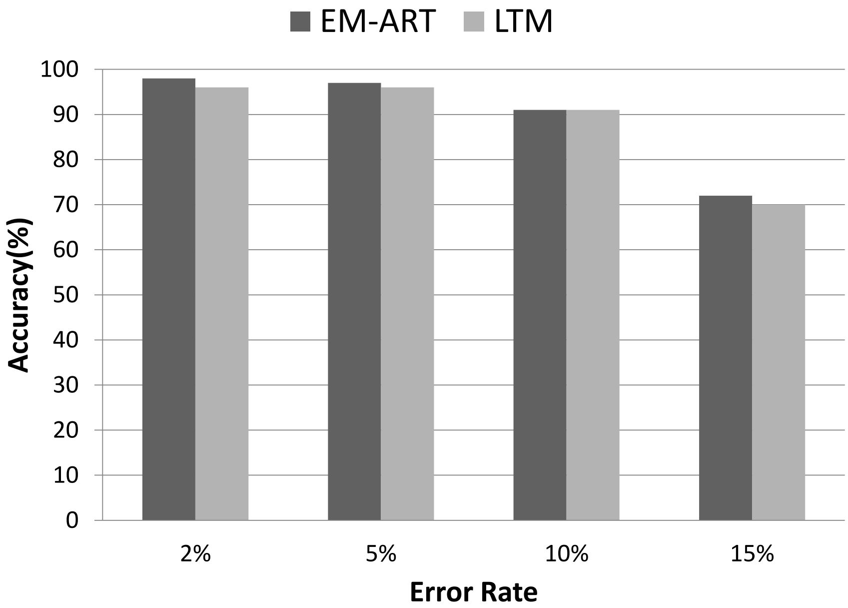


Figure 3.8: Comparison of retrieval accuracy (in %) using full-length cues with various error rates on event representation.

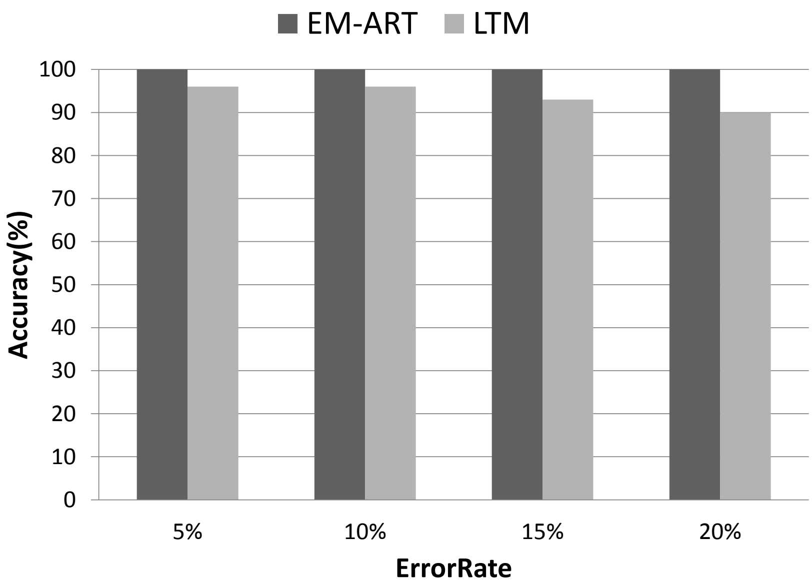


Figure 3.9: Comparison of retrieval accuracy (in %) using full-length cues with various error rates on sequence representation.

are typically subjected to continuous decaying on their memory strength and eventually deleted from episodic memory due to low reactivation frequency; Meanwhile, a consistently happening experience tends to be preserved by high repetitions. The episodic memory model should be able to maintain the actual experiences despite the limited noises present. Therefore, the actual experiences are more likely to be retrieved from memory. In this way, not only forgetting helps episodic memory to maintain a manageable memory size in the long term, but it also enhances the robustness and reliability of the model performance in a noisy environment.

In this section, we simulate four sets of noisy training data as shown in Algorithm 3.6. By setting r in Algorithm 3.6 to 5, 10, 15 and 20, we generate four noisy data sets, each with 77350 events and 1000 episodes. The generated data sets respectively containing 5%, 10%, 15% and 20% errors on their event representation 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.

Algorithm 3.6 Generation of Noisy Training Set

Input: Error rate $r \in (0, 100)$, original data set Set_{orig} and an empty data set Set_r
Output: Data set Set_r
1: Copy Set_{orig} into Set_r
2: Generate a noisy data set from Set_{orig} using Algorithm 3.4 with a passing argument of r
3: Append the generated data set to Set_r
4: Repeat step $1 - 3$ for D for 4 times

We set the initial confidence $s_{init} = 0.5$, 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 $s_{init} = 0.5$, 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 3.5 with comparison to their corresponding models without forgetting. From Table 3.5, we observe that as the error rate increases from 5% towards 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 continuously recognition of and thus deletion of noisy patterns.

Table 3.5: Comparisons of the EM model sizes (in numbers of event and episode nodes) with/out forgetting at various levels of vigilances

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

After the models are built, we have conducted 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; (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 3.4. The performance for these retrieval tests are compared with the counterparts without forgetting.

As shown in Figure 3.10 and 3.11, forgetting helps episodic memory to retrieve more episodes correctly despite the reduction in memory size. The only exception is on the

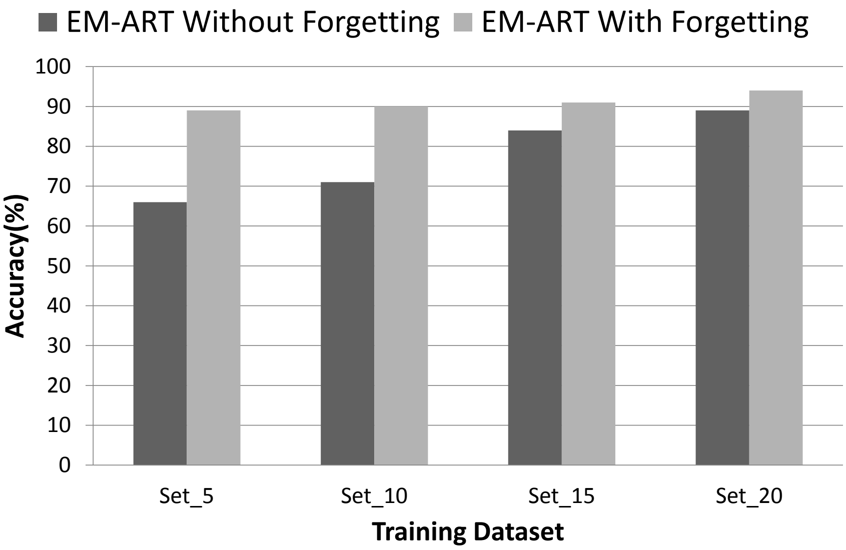


Figure 3.10: Comparison of retrieval accuracies (in %) with 1/3 of episodes from end as partial cues.

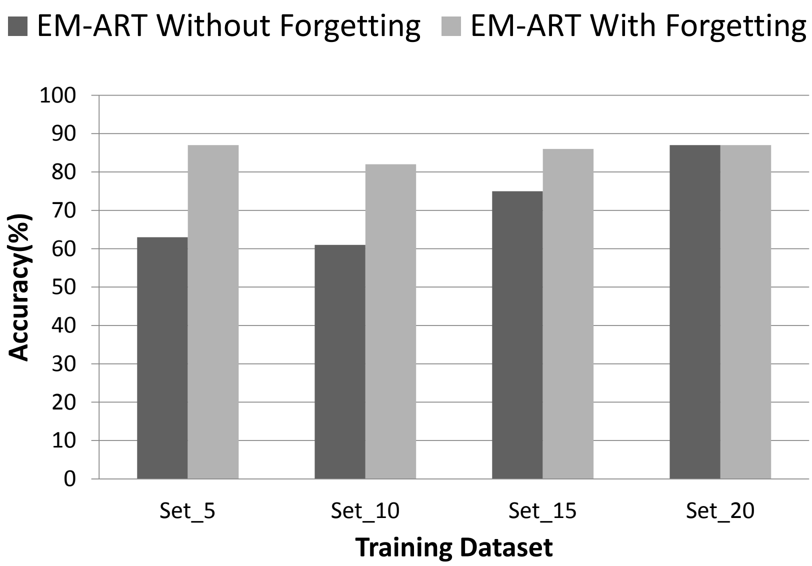


Figure 3.11: Comparison of retrieval accuracies (in %) for with 1/5 of episodes from end as partial cues.

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 distinguished erroneous training samples and that the original experiences can be retrieved more accurately.

3.8 Discussion on 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. [19, 40, 41]). 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 [19] have used statistical methods to deal with imperfect and noisy cues, they consider memory trace as continuous series of events with no coherent representation of chunks of episodes as units of experience. Our model addresses this issue by representing events as multi-channel 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 (Soar) to store episodes instead of the linear trace (e.g. [1]). 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 to store 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 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. [20, 5]). 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 (Adaptive Resonance Theory) neural network with spectral timing encoding to model timed learning in hippocampus [106]. 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 composed of role-entity bindings [107, 2], without considering their spatio-temporal relations. Our 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 [108, 109, 27] 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.

3.9 Summary

We have presented a new episodic memory model called EM-ART, based on a class of self-organizing neural networks known as fusion Adaptive Resonance Theory and the technique of invariance principle. The model encodes the potentially complex conceptual and spatio-temporal relations among past situations. The stored information can be retrieved with various imperfect cues containing noises and errors.

We have conducted empirical experimental evaluation on EM-ART using a first-person shooting game, as well as a word recognition benchmark test. Various tests are performed on the built memory model to access its efficiency of possible memory retrieving during the games. The experimental results show that the model is able to provide a robust level of performance in encoding and recalling events and episodes even with incomplete and noisy cues. This is mainly due to its approximate retrieval using resonance search. 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 is regulated by discarding trivial and noisy information.

Chapter 4

Memory Consolidation: From Episodic Memory to Semantic Memory

In Chapter 3, we present a fusion ART-based neural model that learns episodic memory in response to an incoming stream of events. In this chapter, we propose a semantic memory model to learn general facts and knowledge. An additional knowledge transfer process as reported as [110], wherein the information stored in the episodic memory can be consolidated to produce more general and abstract knowledge in semantic memory. Our experiments show that our model can provide the dual memory system succeeds to provides a meaningful memory consolidation to semantic knowledge. Using the same game domain as the last chapter, we integrate the dual memory system into a situated agent to demonstrate how this memory system may improve its performance. The experimental results show that the memory consolidation and forgetting in the integrated episodic-semantic memory system can jointly improve the retrieval accuracy and the task performance and strike a balance between learning useful knowledge and memory efficiency over time.

The rest of this chapter is organized as follows. Section 4.1 provides some background information on the modeling of knowledge transfer between episodic and semantic memory. Section 4.2 reviews the related neurobiolocal studies, which forms the basis of the proposed dual memory systems. Section 4.3 discusses and compares the existing models of the semantic memory and those of the episodic-semantic dual memory systems. Section 4.4 gives the details of the semantic memory model implemented. Section 4.5 presents the incorporated memory consolidation process, followed by the experimental results shown in Section 4.7. The overall performance of the integrated model is given in Section 4.8. The final section concludes the work presented.

4.1 Introduction

In this chapter, we extend the modeling of the multiple memory systems by augmenting EM-ART with a general procedure wherein the contents of episodic memory may be consolidated and transferred to a more permanent form of semantic memory. With the dual memory system consisting of episodic and semantic memory, a situated agent can recall the content in episodic memory through a memory consolidation process to gradually extract and learn general facts and rules as semantic memory and improve its overall performance. This kind of dual memory mechanism has also been known to prevent memory interferences and catastrophic forgetting [51]. Essentially, in this dual memory system, episodic and semantic memory run and learn independently in parallel but at different paces. Episodic memory serves as a long-term temporary buffer for rapidly storing events and episodes. The stored events and episodes then can be recalled at a later time through a memory consolidation process to gradually extract and learn general facts and rules as semantic memory.

Contrary to many models of declarative memory in which the emphasis is mostly on memory as an information storage with flexible retrieval mechanism [111, 1, 112], some issues that we want to explore are (1) how the contents of episodic memory can be selected through memory consolidation to form the right semantic knowledge; and (2) how the consolidated knowledge can be used to support cognitive functions and task performance. Hence, besides the explanation on the detailed transfer process, we apply this dual memory system into an exemplar agent in the Unreal Tournament domain to demonstrate how the memory system performs the knowledge transfer.

4.2 Neurobiological Studies on Hippocampal-Cortical Interactions

The very first evidence for the consolidation process between episodic and semantic memory comes from the phenomenon called retrograde memory gradients [113, 114], wherein recent memories are actually more impaired by hippocampal damage than more remote ones. The relatively dense loss of recent memories suggests that the learned knowledge in episodic memory is not instantly made permanent and requires a continual consolidation process into a more persistent storage form [115].

In the standard model of the memory consolidation process [116, 117, 89], when the novel information is initially encoded into hippocampus, its corresponding hippocampal representation is bound into the synchronously-activated memory traces in other related structures, including cortex, medial temporal lobes and diencephalon [118]. This initial hippocampal-neocortical binding results into a "weak" knowledge representation in cortical region [119]: the cortical representation cannot be activated though its weak neuronal connections alone, but requires the stimuli inputs from the hippocampus and related structures. After the initial knowledge learning, a process of the memory consolidation then begins, wherein the learned hippocampal information is reactivated through conscious and/or unconscious memory recalls during sleep or some offline processes [120]. As the hippocampal representation is continually reactivated and "teaches" the cortex (i.e. fires the activation of the cortical representation through memory binding) during the consolidation process, the bounded cortical knowledge representation is gradually enhanced [120]. Eventually, the consolidation process ends when the cortex representation is strong enough to sustain the permanent memory trace and mediate its retrieval alone [117]. Therefore, though the memory consolidation, the knowledge originally represented in the hippocampus is gradually transferred to the more permanent storage in cortex[120].

4.3 Related Work

As opposed to episodic memory, semantic memory stores meanings, concepts, rules, and general facts [11] rather than specific experienced events. Taking place all over cortical/neocortical areas of the brain, semantic memory can maintain information more permanently than episodic memory. Various types of structure and representation have been proposed for semantic memory over the past decades. One of the earliest model suggests semantic memory to store simple logical propositions encoded as nodes and links of a *semantic network* [71]. 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 Cognitive Maps (FCM) is also proposed incorporating fuzzy logic for representing the extent of relationship between concepts [22]. FCM can represent causality between concepts supporting causal reasoning. Another model of declarative memory is also suggested to represent general facts as *chunk*s in ACT-R cognitive architecture [23]. A declarative chunk can describe properties of a concept with a sum of activation level reflecting its general usefulness in the past. Besides representing meanings as symbolic conceptual relations, other approaches use statistical methods to store knowledge. Semantic models like Hyperspace Analogue to Language (HAL) [24], Latent Semantic Analysis (LSA) [25], and REM-II [19] maintain the level of statistical co-occurences between memory items. The statistical models can handle partial or degraded retrieval cues by applying statistical inferences.

Besides abstract computational models above, some others are based on neural architectures corresponding to the memory systems in the brain. Hinton [73] proposes one of the first connectionist architecture emulating the semantic network model by setting up interconnected neural fields reflecting different elements of a proposition. Beyond representing relationships between concepts, the connectionist architecture supports recollection and generalization through pattern completion accross the network. Further, Rumelhart [28] also develop a similar connectionist model that can automatically learn relational and hierarchical structure of a semantic network. Using a backpropagation learning model, the connectionist architecture can categorize and discriminate different concepts without direct supervision. Other works also base their models on the actual neurocognitive functionality of neocortex in the brain. Farah and McClelland [26] suggest a bidirectional network model consisting of different interconnected neural fields corresponding to their sensory-functional features. 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 [3]. Different models and structures above indicate that semantic memory is not a monolithic unitary model but may involve multiple representation and learning mechanisms.

Most semantic models mentioned above are still considered as isolated memory systems that process and acquire semantic knowledge directly from the inputs without interacting with another part in the whole memory systems. However, some models also consider episodic memory to be attached with semantic memory as a whole memory system. REM-II [19] connects episodic memory and semantic memory together to learn statistical relationships between items within and accross time. Another episodic memory model based on the SOAR cognitive architecture [1] embeds episodic memory directly to the symbolic semantic memory model as additional properties providing contextual and historical information of each assertion and update in the memory. A distributed approach like TESMECOR [27] also consider episodic memory as distributed neural connections that also support semantic representation. Although the integrated approaches of episodic memory and semantic memory modelling may provide robust mechanisms to store and retrieve knowledge based on both temporal and relational structures, they still do not reflect the current existing neuropsychological evidences of memory.

The cortico-hippocampal neural model [5] is made to explain why episodic memory and semantic memory are separated in the brain. The model 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 encode the input and the recalled patterns. On the other hand, the neocortex learns and categorizes the input based on the internal representation formed by the hippocampus. In this way, episodic memory and semantic memory can work together to process abstract categorization while they can accurately retrieve specific information. A more realistic model of episodic-semantic memory interaction called Complementary Learning Systems (CLS) [51] reflects the network structure and connections between hippocampus and neocortex in the brain and comprises a particular memory consolidation process. Based on neuroscientific evidences that neurons in hippocampus are reactivated spontaneously during slow wave sleep [76] 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. The model also incorporates a forgetting mechanism of hippocampus in which the strength of neurons are decayed over time before reinstated during or beyond the consolidation process.

Although the complementary neocortex and hippocampus models above can be conformed with the real evidences of memory consolidations and lesions behavior in the brain, they still omit the sequential or temporal processing in hippocampus. In this research, we follow the complementary model of episodic memory and semantic memory in which both memory systems learn separately and independently but the formation of knowledge in semantic memory is also supported through the consolidation process from episodic memory to semantic memory. However, we also consider that the sequential order learnt in episodic memory is also a crucial factor enabling the complex structure formation in semantic memory beyond pattern generalization and discrimination.

4.4 Semantic Memory Model

Different from episodic memory, we view that the semantic memory is not unitary. In other words, there may be different types of semantic memory network, each represents different structure of knowledge. In contrast to episodic memory, each entry in semantic memory generalizes similar inputs into the same category rather than as separate entries. Each type of semantic memory can be made as a fusion ART with each input field representing a property or an attribute of a concept. The generalization can be achieved by lowering the vigilance parameter ρ so that slightly different input patterns will still activate the same category.

Figure 4.1 illustrates the structure of various types of the semantic memory. A semantic memory network may consist of domain specific associative rules (e.g a set of association between a certain object and its location in the environment, a set of rule associating the effectiveness of a certain weapon and the distance to the opponent) or generic causal relations associating a particular type of event to another that follows. These types of semantic knowledge can be derived by exposing the played back items from the episodic memory to the input of the semantic memory using a lower vigilance parameter and a smaller learning rate such that similar instances may gradually be clustered together regardless of their order.

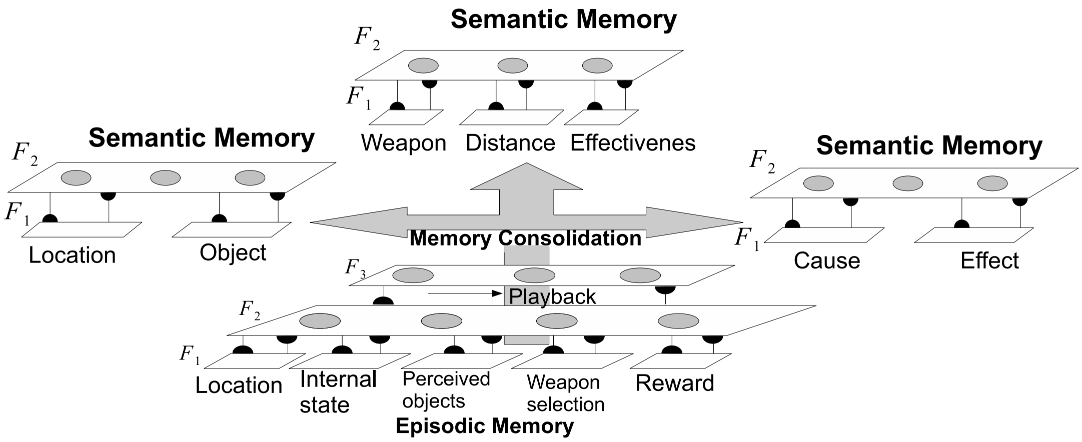


Figure 4.1: Different types of semantic memory in UT domain and their memory consolidation process with episodic memory

4.5 General Process of Knowledge Transfer from Episodic Memory to Semantic Memory

Besides encoding and storing events sequentially, the other functionality of episodic memory is to consolidate the stored information into permanent abstract (semantic) forms [12]. The process of transferring episodic memory items into permanent knowledge in semantic memory is not a unitary process and may involve different parts and areas in the brain. An existing model of episodic memory, based on a well known cognitive architecture (Soar), has involved the formation of more abstract knowledge from episodic memory [1]. A neural network model has also been formulated to extract abstract knowledge involving dynamic variable bindings from a sequential trace of events [121]. Each of these models requires specific mechanisms to extract particular structures of knowledge. Several mechanisms and memory structures are also applied in EM-ART to extract more general and abstract knowledge based on the nature of the semantic knowledge to be extracted. Figure 4.2 illustrates the memory system supporting the transfer of information from EM-ART to a semantic memory structure.

During a process of knowledge transfer, EM-ART reproduces each stored episode as

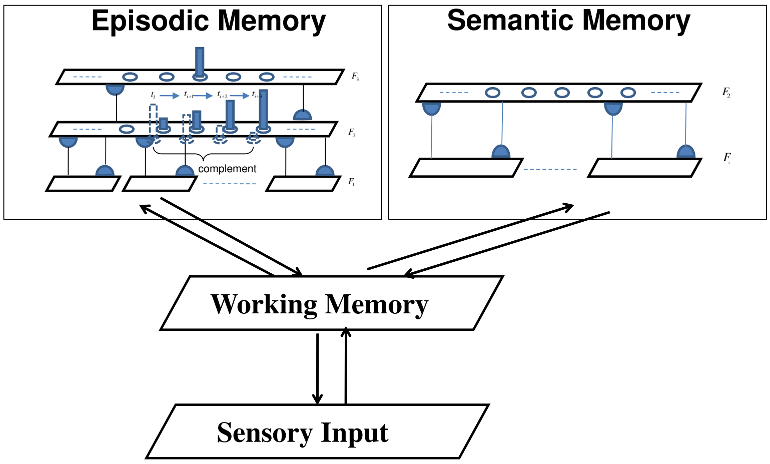


Figure 4.2: An illustration of the knowledge transfer process from episodic to semantic memory

a memory playback process as described in Section 3.4.2. During each episode reproduction, each associated event in the episode will be presented as both the episodic memory output (via F_1) and the activation pattern of the working memory. The ordering of event reproductions should be consistent with the temporal information stored (via weights of F_2). As each event is presented, it will be reevaluated and checked against its relevance with the current knowledge transfer process. In case the presented event describes the experience of interest, the event representation (shown in Figure 3.2) held in the working memory is forwarded to semantic memory as a training sample for learning the specific semantic knowledge. Otherwise, the content of working memory is discarded and the reproduction is continued from the next stored event.

4.6 An Example on Knowledge Transfer

To understand the knowledge transfer process between episodic and semantic memory, we consider an autonomous agent playing the first-person shooting games of UT.

Example: An agent plays the UT game and learns its own experience as episodic memory. Given an event representation illustrated in Figure 3.2, a sample episode/battle learned by the agent can be described as a ordered list of occurred events shown below.

E_1[1]: (*location* = (x_1, y_1, z_1), *health* = 95[2], *ammo* = 50[3], *reachable item* = *true*, ..., *running around* = *true*, ..., *reward* = 0.5[4])

E_2: (*location* = (x_2, y_2, z_2), *health* = 95, *ammo* = 50, *reachable item* = *true*, ..., *collecting item* = *true*, ..., *reward* = 0.5)

E_3: (*location* = (x_2, y_3, z_3), *health* = 100, *ammo* = 50, *pickup small health* = *true*, ..., *collecting item* = *true*, ..., *reward* = 0.5)

...

E_n: (..., *enemy distance* = 1000[5], *is shooting* = *true,* selected weapon=ASSAULT RIFFLE..., *engaging in battle* = *true*, ..., reward= 0.5)

E_{n+1}: (..., *enemy distance* = 500, *is shooting* = *true,* selected weapon=ASSAULT RIFFLE..., *engaging in battle* = *true*, ..., *reward* = 0.6)

E_{n+2}: (..., *enemy distance* = 100, *is shooting* = *true,* selected weapon=ASSAULT RIFFLE..., *engaging in battle* = *true*, ..., *reward* = 0.9)

E_{n+3}: (..., *enemy distance* = 100, *is shooting* = *true,* selected weapon=ASSAULT RIFFLE..., *engaging in battle* = *true*, ..., *reward* = 1)

This episode describes a battle, wherein the agent collects items in the initial stage, engages with the opponent and finally wins the game. We assume that the agent is embedded with a semantic memory model to learn the weapon effectiveness. As presented

[1] E_n refers to the n-th event within an episode
[2] the value of the *health* is bounded to [0,100]
[3] the value of the *ammo* is bounded to [0,100]
[4] the reward value of 0.5 stands for no battle outcome received at the current state; a reward value in the range of $(0, 0.5)$ and $(0.5, 1)$ refers to be damaged by and damage the opponent respectively, and the reward value is proportionated with the amount of damage incurred; the reward of 0 or 1 refers to a event of being killed or killing
[5] a visible enemy distance is between 0 to 3000

in Section 4.4, the semantic memory model takes a rule-based representation. Each semantic rule in this model can be written as "IF *selected weapon* is a, AND *enemy distance* is $[d_1, d_2]$, THEN the *damage caused* is usually in the range of $[r_1, r_2]$". Assuming the agent has no prior knowledge on weapon effectiveness, the semantic memory model is initialized to contain no rule. This is to indicate that the agent has no prior knowledge on the weapon effectiveness.

At a certain point of time, the process of knowledge transfer from episodic to semantic memory is initialized. The episode described above is replayed wherein each of its events, from E_1 to E_{n+3}, will be read out one by one to the working memory according to the order of occurrence. Once an event is presented to the working memory, the transfer process evaluates its relevance to the current semantic memory learning precess. To order to learn the semantic knowledge on the weapon effectiveness, an event is considered to be relevant to the learning process if its *is shooting* flag/attribute is set in its corresponding event representation. Hence, in the sample episode, the event representations of E_n to E_{n+3} are fed to the semantic memory as the training samples to learn the semantic rules on weapon effectiveness. Each training sample is a 3-tuple of (*selected weapon*, *enemy distance*, *reward*).

As the agent gradually learns the semantic knowledge on weapon effectiveness from its battle experiences, the agent gradually utilized its semantic knowledge by making the memory query for the best weapon to choose while deciding to engage in a battle. The query can be in the form of "which weapon can cause the greatest damage if my distance to the enemy is x?" If the query is returned with a optimal weapon choice, the agent improve its decision making through a more sophisticated scheme of weapon selection. The knowledge transfer process can be easily extended to learn other types of semantic knowledge in two steps: (1) determine the network structure of the semantic knowledge to be learned; (2) define the process of relevance check on episodic events. The following two sections provide more implementation details on the knowledge transfer process.

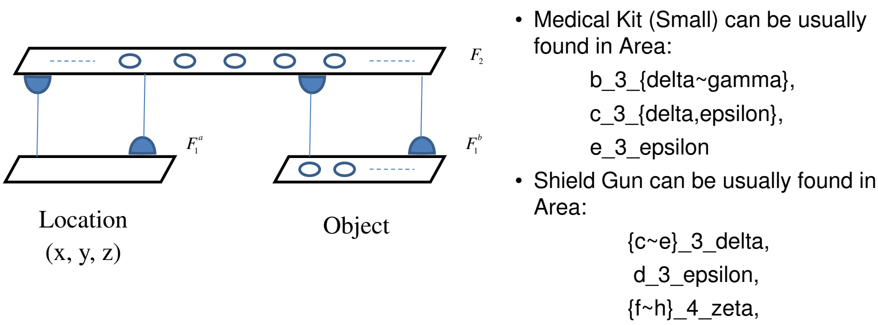
4.7 Illustrations on Knowledge Transfer

In our experiment domain, an agent in UT can develop a wide range of semantic knowledge from its battling experience to enhance its performance. We identify the following two types of knowledge of agent's interest: (1) Association rules between collectable items (i.e. weapons and medical kits) and their collecting locations on the battlefield; (2) Causal relation between weapon used, distance to enemy, and damage caused by the weapon firing. We implement these knowledge transfer and investigate the extracted semantic knowledge as a result of the transfer.

4.7.1 Learning of Association Rules

In the UT games, the agent usually can make better decisions if it has a correct estimation on where are the collectable items are placed. This can be expressed as the association rules between the items and their collecting locations. We design the fusion-ART based memory for these groups of association knowledge as shown in Figure 4.3(a). The Semantic Memory (SM) model consist of two F_1 fields: the Location field represents the battlefield coordinates while the Object field holds exactly one semantic representation for each of the available collectable items. Each F_2 code represents an association rule in the form of "At location (x, y, z), the item a is usually present".

During the knowledge transfer as described in Section 4.5, each event stored in EM-ART is replayed and the events describing an encounter with collectable items are selected. To learn the association rules between the items and their collecting locations, each selected event together with the agent's location and items encountered are copied to the Location and Object fields in the SM model respectively as a training sample for learning the association rules. In the experiment, we investigated a total of 100 battles containing 3602 events. During the transfer process, 225 events are identified to involve the encountering of collectable items. These 255 events are used to build a SM model



(a) Semantic memory of association rules (b) Sample assoication rules

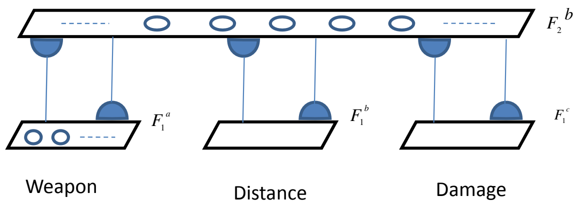
Figure 4.3: Learning of association rules

containing 54 association rules. Figure 4.3(b) shows six sample rules taken from the 54 learned rules. As shown in Figure 4.3(b), an agent can access the most-likely locations of the items and pick up essential items in the closest location by querying this SM.

4.7.2 Learning of Causal Relations

Generally, the agent performance in the battle can be improved if it has a better understanding on the characteristics of each available weapon. We define these characteristics of weapons as the causal relations between weapon, distance to enemy and damage caused. Therefore, we present the fusion-ART based memory for this type of knowledge as shown in Figure 4.4(a). The SM model has three F_1 fields: the Weapon field holds exactly one semantic representation for each weapon in the game; Distance field shows the distance between the agent and its enemy at the time of firing; Damage field gives the resulted damage to the enemy (reduction of enemy' health level) caused by the firing. Each F_2 code represents a learned causal relation in the form of "IF the weapon is a, and distance is $[d_1, d_2]$, the caused damage is usually in the range of $[r_1, r_2]$".

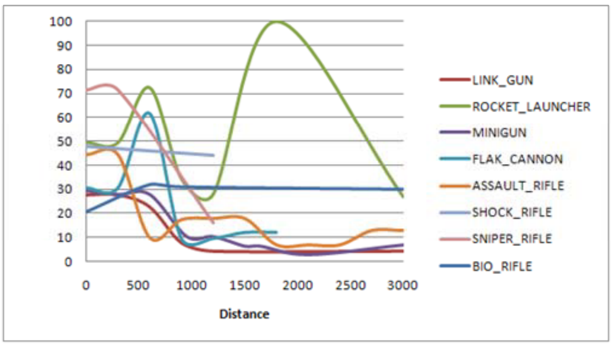
During the knowledge transfer, the events which involve weapon firing are selected to learn the causal relations of "Weapon + Distance $\rightarrow$ Damage". For each selected



IF Weapon = FLAK_CANNON,
Distance = [1143 ~ 1199],
THEN Efficiency = [0 ~ 2].

IF Weapon = FLAK_CANNON,
Distance = 600,
THEN Efficiency = [58-64].

(a) Semantic memory of causal relations (b) Sample causal relations



Weapon	Effective Distance	Max Distance
SHOCK_RIFLE	1500	4000
ROCKET_LAUNCHER	1900	20000
SNIPER_RIFLE	2000	20000
MINIGUN	600	4000
FLAK_CANNON	500	1500
BIO_RIFLE	1400	2000
LINK_GUN	1200	5000
ASSUALT_RIFLE	2000	300

(c) Leaned damage expectation (d) Game-predefined damage by various weapons and distances

Figure 4.4: Learning of causal relations: weapon + distance $\rightarrow$ damage

event, the current weapon held by the agent, the distance to the enemy and the resulting damage (e.g. reward) are copied to the Weapon, Distance and Damage fields in the SM model respectively as one-time input activation for learning. In the experiment, we investigate a total of 200 battles containing 12130 events. During the transfer process, 1894 of the events include weapon firings which lead to 37 causal relations created in the SM model. Figure 4.4(b) gives two sample relations taken from the 37 learned rules. We further plotted the knowledge of these rules into damage prediction curves against distance for different weapons, as shown in Figure 4.4(c). The learned weapon curves can be compared and verified by the game-defined weapon characteristics illustrated by Figure 4.4(d). Comparing Figure 4.4(c) and (d), for frequently used weapon, e.g. Minigun, the damage prediction from SM are roughly consistent with the pre-defined weapon features in the game; however, for the rarely used weapon, e.g. Bio-Rifle, there are large differences between the learned knowledge and the game-set weapon rules. This is mainly due to the insufficient learning with the limited examples of using these weapons. With adequate battle experiences, we can expect a more accurate weapon damage prediction, even compared with the game-set weapons rules, which omit all case-dependant practical considerations.

4.8 Experiments on Integrated Agent

In order to test the integrated dual memory model, we embed the episodic-semantic memory system into an autonomous non-player character (NPC) agent in the Unreal Tournament (UT) game [122].

The UT game is a commercial first-person shooting game, which requires the real-time decision making in the complex environment. To be competitive in the UT, the agent needs the expertise and domain specific knowledge and skills on this domain. One promising advantage of the UT game domain is that it provides the easy and standard mechanics to implement an self-defined agent playing the game through Pogamut [123]. Pogmut is an free Integrated Development Environment (IDE) which manages the communication between the game and the built agent. Pogmut also provides the templates to build various types of virtual agents simply by overriding some required methods. With the ease of implementation and comparison, some recent works [124, 125, 126, 127] have presented various types of behavior learning model in the domain of UT game. In [128, 129, 130], groups of virtual UT agents have been proposed and implemented in order to study the process of the multi-agent decision making. Figure 5.8 gives a screen snapshot from the game play in UT.

Figure 4.5: A screen snapshot of Unreal Tournament

The experiments in UT are conducted to validate our memory model, which should produce useful knowledge for the agent that improves its performance. The knowledge is not pre-wired but learned by the agent while it explores and engages in battles inside the game. We also investigate whether the forgetting process may sacrifice the agent's performance.

The scenario of the game used in the experiment is "Death match". 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.

4.8.1 The Baseline Agents for Comparison

All agents that we evaluate in the experiment play against an NPC agent called *AdvanceBot* that behave according to hardcoded rules. There are four different hard-coded behavior modes in *AdvanceBot*: (1) Running around behaviour, in which the agent will run around exploring the environment randomly; (2) Collecting items behavior, in which the agent will go and pick up collectible items; (3) Escaping from the battle situation, in which the agent will turn and run away from the opponent; (4) Engaging in battle, in which the agent will approach its opponent and shoot to kill it.

AdvanceBot always chooses one of the four behaviors based on a set of predefined rules. Under the battle engagement behavior, the agent also always tries to select the best weapon available for shooting. The weapon selection rules are based on some heuristics optimized for a certain environment map used in the game. As a performance comparison, another agent (name it *RLBot*) is made to employ the same set of behaviors but its selection is conducted dynamically based on a fusion ART neural network conducting reinforcement learning algorithm.

The state, action, and reward vectors in Figure 4.6 correspond to the input fields in a multi-channel ART network of *RLBot*. 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 field and the reward (set to the maximum), the network searches the best match category node and reads out the output to the action field indicating the behavior type to be selected. The network then receives feedbacks in terms of the new state and reward (if any).

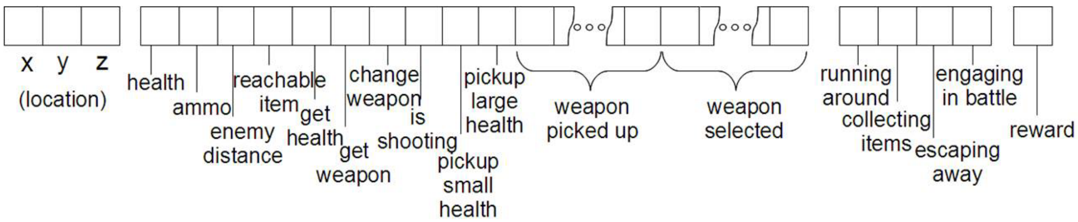


Figure 4.6: The input vectors of state, action, and reward to *RLBot* and *RLBot++* in UT

The network learns by updating the weighted connections according to the feedback received and applying temporal difference methods [131] to update the reward field if the immediate reward is absent. The agent receives the reward signal (positive or negative) whenever it kills or is killed by another agent. 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*.

4.8.2 Episodic-Semantic Memory Enhanced Agent

The model is embedded in an agent with the same architecture as *RLBot* but with the episodic memory module 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 4.6. 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 (ρ_e) and the F_2 field (ρ_s) 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. The values of the other parameters remains the same as Section 3.7.1

According to the general process of knowledge transfer presented in Section 4.5, at a certain period of time, the contents of the episodic memory is played back by reading out the events to the working memory. The reinstatement occurs in the period between different battles wherein one agent has just been killed and starting to re-spawn in another place. The semantic memory then acquires the knowledge by learning from the recalled events. In the experiment, only one type of semantic memory about weapon effectiveness is used given the distance towards the enemy. Whenever the value of the reward field in the event vector is large enough to be considered as a successful killing (0.5 is the threshold), the values of weapon selected, opponent distance, and reward (or the effectiveness to kill) fields are fed and learnt by the semantic memory.

The multi-channel ART network of the semantic memory used in the experiment to learn the causal relation between weapon, distance to enemy and weapon effectiveness as discussed in Section 4.7.2. 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 experiment also uses forgetting in episodic memory with the threshold (t_s) and reinforce rate (r_s) set to 0.0001 and 0.5 respectively. To evaluate the effect of forgetting, different decay rates (δ_s) in the events field F_2 are used: 0 (no forgetting), 0.005, 0.01, and 0.02.

4.8.3 Results and Discussion

Experiments are conducted by letting *RLBot*, *RLBot++* and the memory-based *RLBot* (called *MemBot*) with different forgetting decay rates (δ_s=0.005, δ_s=0.01, and δ_s=0.02) to individually play against *AdvanceBot*. For practical reason, *MemBot* without forgetting (δ_s=0) is excluded from performance comparison as the program overloads the system memory soon after the game starts causing the system to halt and the agent refrains from playing. A single experiment run consists of 25 games or trials, which is counted whenever the agent kills or is killed by another agent.

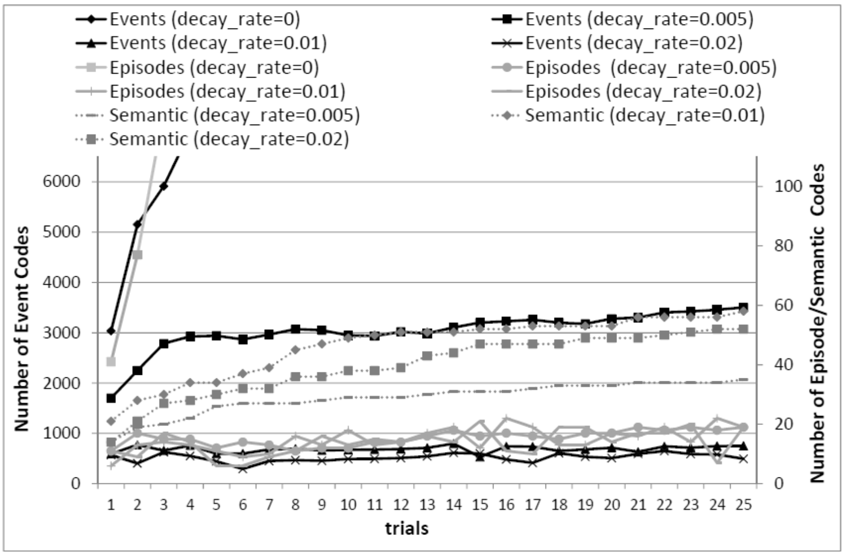


Figure 4.7: Memory usage for events, episodes, and transferred semantic knowledge with different forgetting decay rate during the game play

Figure 4.7 shows the memory size taken up in the episodic memory (in terms of the number of nodes in F_2 and F_3 of a *MemBot*) and the number of nodes created in the semantic memory with different δ_s in F_2 sampled from a single run against *AdvanceBot*. Without forgetting ($\delta_s = 0$), the memory space is taken up rapidly into its limit after about three trials.

In contrast, the forgetting mechanism can make the memory size converge and stabilize at certain points. Hence the agent can always perform and learn continuously. It is clearly shown that the larger the decay rate, the smaller number of codes (nodes) is produced in episodic memory. Interestingly, in semantic memory, a low decay rate (e.g δ_s=0.005) creates lesser codes comparing with those obtained with higher decay rates (e.g δ_s=0.01 and δ_s=0.02).

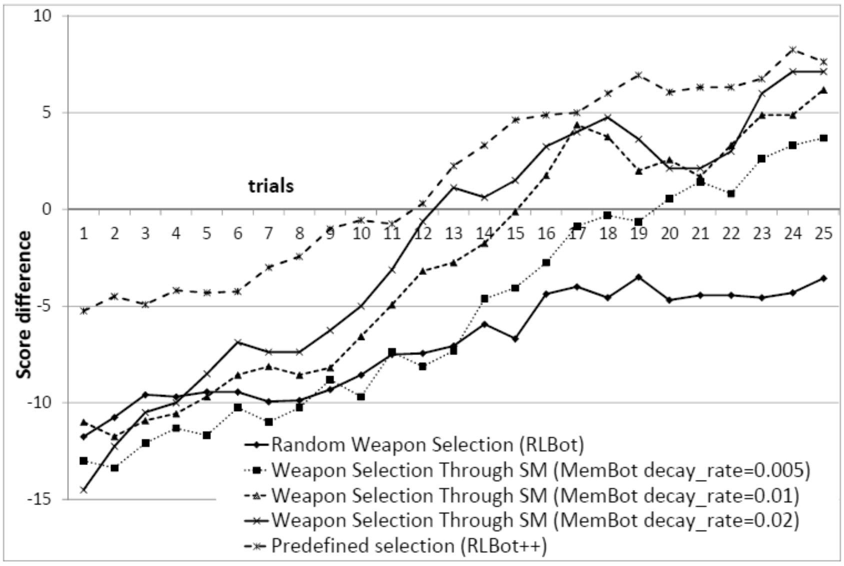


Figure 4.8: Performance of *RLBot*, *RLBot++*, and *MemBot* over 25 trials

Figure 4.8 plots the performance of both *RLBot*, *RLBot++* and *MemBot* with different δ_s in terms of game score differences against *AdvanceBot* averaged over four independent runs. It shows that incorporating the episodic and semantic memory model improves the learning which results in a much better performance than using the reinforcement learning alone. This indicates that the semantic memory successfully learns useful knowledge for the weapon selection portion of the reasoning mechanism. Surprisingly, the results also indicate that with a larger forgetting rate (e.g δ_s=0.01 and δ_s=0.02), the performance and learning efficiency are better than those obtained with the smaller one (δ_s=0.005) and can eventually reach the performance using the optimized rules model. In other words, forgetting less important things faster can make learning better. One explanation of this beneficial effect of forgetting is that noisy events that could impair the consolidated knowledge are filtered before being generalized in semantic memory. The semantic memory would thus end up with the appropriate generalization and some specific but necessary information.

4.9 Summary

We have presented the integrated episodic-semantic memory model for virtual agents integrating two separate modules of episodic and semantic memory. The stored contents of episodic memory can be recalled to derive abstract knowledge and general facts into more permanent forms in semantic memory. The episodic and semantic memory modules are realized with fusion ART neural networks as two independent but connected networks operating in different paces of learning. A forgetting mechanism is also applied to regulate the size of memory by removing insignificant entries.

Our experiments confirm that an explicit memory model can improve the agent learning and performance by acquiring useful knowledge for the task at hand through memory consolidation, relieving the agent from continuously reasoning and processing the information for learning. It is also demonstrated that the forgetting regulates the memory size while the performance is still improving. Moreover, the experiment shows faster forgetting can even result in better learning. This indicates that the forgetting can successfully filter out insignificant entries while maintaining the useful ones. The findings can inspire the exploration of forgetting as a useful feature of intelligent agents and machine learning systems in general.

Chapter 5

Declarative-Procedural Memory Interactions

It has been well recognized that human makes use of both declarative memory and procedural memory during the process decision making and problem solving. In this study, we present our investigation into the constraints and principles for modeling the interactions between the declarative and procedural memory modules. To this end, we propose a computational model with the overall architecture and individual processes for realizing the interaction between the declarative and procedural memory based on self-organizing neural networks. We formalize two major types of memory interactions and show how each of them can be embedded into autonomous reinforcement learning agents. Our experiments based on the Toad and Frog puzzle and a strategic game known as Starcraft Broodwar have shown that the cooperative interaction between declarative and procedural memory can lead to significant improvement in task performance. In addition, semantic knowledge may be transferred to procedural memory through a natural learning process.

5.1 Introduction

Human brains have been well recognized as multiple memory systems [9] consisting of notably declarative memory and procedural memory. Declarative memory is an explicit record of what we encounter and what we learn. Procedural memory, on the other hand, refers to the implicit memory of skills and reflex responses, wherein the knowledge is usually difficult to articulate or explain. While procedural learning is essential to the development and utilization of both motor and cognitive skills, declarative memory represents high level concepts and knowledge which forms the basis of our understanding and provides guidance to reasoning and decision making.

Declarative and procedural memory do not function independently. In view that memory interaction is an integral element of human cognition, this research reports our investigation into the interaction between declarative and procedural memory. To this end, we present a cognitive model with an explicit modeling of procedural and declarative memory. The architecture consists of four main components, namely, the working memory module, the procedural memory module, the declarative memory module and the intentional module. Each of the memory modules in our system is built based on self-organizing neural network models known as fusion ART [34]. As a generalization of Adaptive Resoannce Theory (ART), the multi-channel network provides a set of universal computational processes for encoding, recognition, and reproduction of patterns. Previous works [132, 133] have used fusion ART as a building block for various types of memory systems, including episodic memory, semantic memory and procedural memory. In this work, we conduct an in-depth study into how the interaction among declarative and procedural memory systems enables the model to produce a more versatile capability in decision making and problem solving. Specifically, we identify and formalize two major types of memory interaction and knowledge transfer processes between semantic memory and procedural memory. More importantly, we show how each type of the memory interaction can be embedded into a reinforcement learning agent.

We have conducted the experiments on two problem domains: (1) the Toad and Frog puzzle and (2) a strategic game known as Starcraft Broodwar. Our experimental results show that the model is able to learn procedural knowledge for the various tasks in the game domains based on reinforcement learning signals from the environment. More importantly, the results show that the interaction between declarative knowledge and procedural skills can lead to a significant improvement in both learning efficiency and performance.

The rest of this research is organized as follows. Section 5.2 briefly reviews recent studies on memory interaction and the related works on memory modeling in cognitive architecture. Section 5.3 presents the framework of the overall architecture and its main components. Section 5.4 presents a formalization of the memory interaction processes among the dual-memory system. Section 5.5 describes the underlying implementation of the individual memory modules. Sections 5.6 and 5.7 present the experimental results. The final section concludes and highlights our future work.

5.2 Related Work

5.2.1 Memory Interaction

Declarative and procedural memory have been shown to interact during problem solving. Recent neurobiological studies [134] on human multiple memory systems have shown that, given a specific memory task, the various memory systems compete for activation with each other throughout the process of task learning. The studies via fMRI further identify that activity in regions associated with declarative memory was negatively correlated with that in the ones associated with procedural memory. Specifically, the activity is initially observed in the declarative memory regions, but declines gradually, while the brain regions of procedural memory gain more activations over time. This activation pattern suggests that, subjects at first try to learn the task based on inferencing and reasoning using declarative memory; however, in the later stage, as the procedural memory learns the stimulus-response association, it is able to make quick decision on its own. Additionally, the research in [135] points to a mechanism in our brains, wherein knowledge can be transferred or transformed from declarative to procedural memory.

For memory interaction, both procedural and declarative memory systems have intensive neural connections with the working memory regions. Working memory refers to the short-term memory system that keeps all the necessary information and knowledge online for use in performing the current task. Since the working memory is strictly subjected to both capacity and time limits [136], during the process of decision making, working memory requires constant interactions with the long-term memory systems to bring the most important information and knowledge into attention. Researchers also found that the novel relations between known concepts must be brought together into the working memory to form novel long-term knowledge [136]. However, further studies have pointed to the parallel processing between the long-term memory system and the working memory [137]. In particular, while the working memory maintains the important information online for solving the problem in hand, the long-term memory systems continuously record and learn these experiences into more permanent forms of knowledge for later use.

5.2.2 Memory Modeling in Cognitive Models

Memory has been an important component of many cognitive models. The Adaptive Control of Thought - Rational (a.k.a. ACT-R) [23] has modelled the declarative memory as a set of competitive learning chunks to hold the current set of known facts, concepts and goals. Procedural memory, on the other hand, is modelled by a central processing system with a set of if/then production rules. At any point of time, ACT-R selects and fires a single production rule by matching its pre-conditions with the pool of available declarative knowledge.

Similar to ACT-R, Soar [83] models the process of problem solving as a complex set of goals, states and actions. Driven by a specific current goal, Soar selects an appropriate action, namely *operator*, from a set of if/then production rules. Different from ACT-R, Soar allows multiple production rules to fire in parallel so that all the relevant procedural knowledge are taken into consideration before make a decision. Soar then selects a single operator to take after rule evaluation. In the case that the available procedural knowledge is insufficient to select an operator, declarative memory retrieves the past similar experiences stored to further guide the rule evaluation. Once the decision is made, a new procedural rule is encoded through *chunking* to deal with similar situations in the future.

CLARION [6] organizes its procedural and declarative memory into two separated subsystems, namely the action-centered subsystem (ACS) and the non-action-centered subsystem (NACS). Their interactions are via the complete control of declarative memory by procedural memory, such that declarative memory provides additional information and reasoning required by the procedural-memory-controlled decision making process. CLARION further suggests an additional supervisory subsystem, namely Meta-Cognitive Subsystem (MCS) to monitor and control the interactions between the two long-term memory systems.

In general, most of the existing works on multiple memory systems focus on a single form of interaction, wherein the declarative memory provides the necessary reasoning to activate the relevant procedural knowledge. In view that human benefits from the existence and interplay of various memory systems, it is the motivation of our work to conduct a deeper study into the mechanism of memory interaction and investigate how it may be used in designing learning agents.

5.3 The Overall Architecture

In this research, we present a neural network-based cognitive model with an explicit modeling of both procedural and declarative memory systems. The architecture contains the minimal structure just sufficient to illustrate the encoding and interactions between semantic memory and procedural memory. This dual memory system may in turn form an integral part of a cognitive agent. As shown in Figure 5.1, the model contains four 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. It also incorporates an attentional mechanism which ensures that the information and knowledge important to the current task is always retained in the memory buffer.
- **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. These tasks are usually automatically performed by procedural memory under the level of consciousness and without the involvement of the intentional module.
- **Semantic Memory** is the storage for meanings, concepts, rules, and general facts about the world, with no relation to specific experiences. The basic forms of semantic knowledge include concept hierarchy, causal relations and association rules.

In this architecture, each of long-term memory modules is capable of performing its own basic operations, including encoding, learning, consolidation and retrieval, independently from all other components. However, the overall decision process is a result of

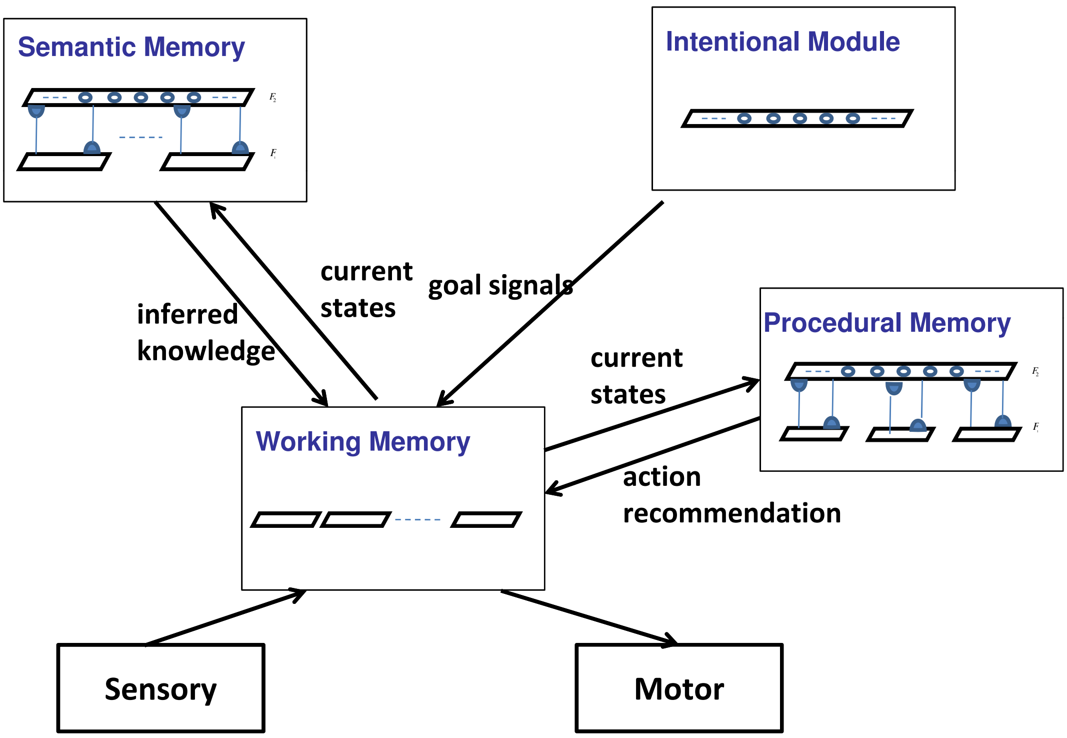


Figure 5.1: The proposed framework of the dual-memory cognitive model.

the complex interactions. In contrast to most cognitive models, the architecture involves no centralized executive control to direct and manage the various components and the learning process. Instead, the interactions among different modules are fully based on the neural connections and the activity propagation mechanism.

5.4 Processes for Memory Interaction

In this research, 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 long-term memory, as follows.

Semantic Memory, denoted by $S = \{S_1, S_2, ...\}$, can be viewed as a set of semantic fragment or rules, where each semantic rule S_i, can be one of the three basic types described as follows: (1) 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."); (2) 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."); (3) 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."). These three types of semantic memory contains the minimal set of semantic knowledge just sufficient from the perspective of the evaluation domains.

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 $P_k : s \rightarrow (a, r)$. Typically through reinforcement learning, procedural memory learns the association of the current state and the chosen action to the estimated reward. The generic flow of the reinforcement learning is presented as Algorithm 5.1 below.

Algorithm 5.1 Reinforcement Learning with 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 $P_{new} : s \rightarrow (a,\mathrm{Q(s,a)})$
    update the current state by $s = s'$
until goal is realized or s is a terminal state

Together, the two memory systems constitute the knowledge base of the cognitive architecture, which guides the behaviors of the agent through their interactions. We present two main types of memory interaction below.

5.4.1 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 5.1 (SP Interaction). *Given the current state s, a threshold τ, and the fol-*

lowing knowledge fragments from semantic memory and procedural memory:

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

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 in Algorithm 5.2.

5.4.2 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 5.2 (PS Interaction)**.** *Given the current state s and a procedural rule*

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

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

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

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

Algorithm 5.2 Reinforcement Learning with Semantic-Procedural Interaction

initialize the Procedural Memory
repeat
  sense the environment and update current state s in Working Memory
  repeat
    Semantic Memory retrieves the most relevant semantic chunk in the form of $S_i : s \rightarrow s'$ or $S_i = (s, s')$
    updates Working Memory by $s \leftarrow s'$
    Procedural Memory searches for a procedural rule $P_k : s' \rightarrow (a, r)$ based on updated current state 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 $P_{new} : s \rightarrow$ (a,Q(s,a))
  update the current state by $s = s'$
until goal is realized or s is a terminal state

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 $P_{new} : s \rightarrow (a, r)$. This interaction and transfer process can be embedded into a reinforcement learning algorithm presented in Algorithm 5.3.

Algorithm 5.3 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' → (a, r)
        Each unmatched attribute value v in P_k causes an increase of its attention level
        in Working Memory
        Semantic Memory searches for a semantic rule S_i : s → s' or S_i = (s, s') such
        that s' contains the missing attribute v and updates the Working Memory
    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 → (a,Q(s,a))
    update the current state by s = s'
until goal is realized or s is a terminal state
```

5.5 Detailed Implementation

The current implementation of the multiple memory model is based on fusion Adaptive Resonance Theory (ART) [34] which categorizes input patterns in a self organizing manner. Figure 5.2 illustrates the fusion ART architecture, which may be viewed as an ART network with multiple input fields. ART employs bi-directional processes of categorization and prediction to find the best matching category. It also learns continuously by updating the weights of neural connections at the end of each search cycle. Fusion ART may also grow dynamically by allocating a new category node if no match can be found. This type of neural network is chosen as the building block of our memory model as it enables continuous formation of memory with adjustable vigilance of categorization to control the growth of the network and the level of generalization. By applying fuzzy operations and *complement coding* [34], fusion ART can also generalize input patterns dynamically and capture a range of values every time it learns.

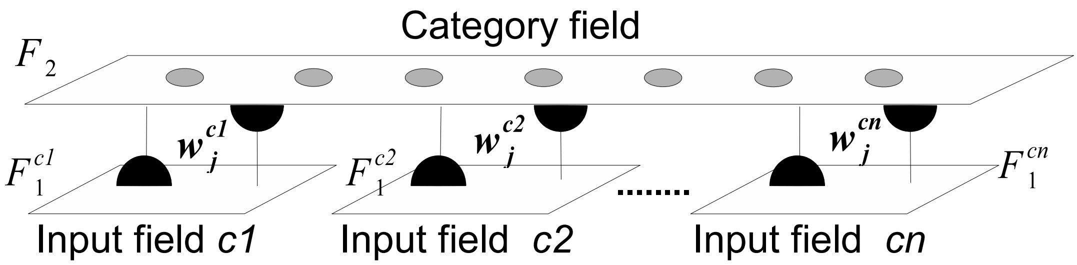


Figure 5.2: The fusion ART Architecture

5.5.1 Working Memory Model

In this architecture, the working memory module is modelled as a vector, denoted by $\mathbf{s} = \{s_1, s_2, ...\}$, which holds all the available information and knowledge about the current situations, including those immediate knowledge retrieved from the long-term memory modules. Functioning as the buffer within the entire cognitive architecture, the working memory shares the knowledge stored among the other modules to perform the subsequent operations, with the aim of accomplishing the current task in hand. Additionally, to evaluate the importance of each piece of information held, each attribute s_i in working memory is associated with an attention factor, a_i. The a_i is initialized with the value of 0, and updated at each point of time such that $a_i(t) = |v_i(t) - v_i^e(t)| \cdot (1 - a_i(t - 1)) + (1 - \sigma) \cdot a_i(t - 1)$, where $v_i(t) \in [0, 1]$ and $v_i^e(t) \in [0, 1]$ are the actual and expected value of the attribute s_i at time t and σ is the decay rate for the attention factors. As

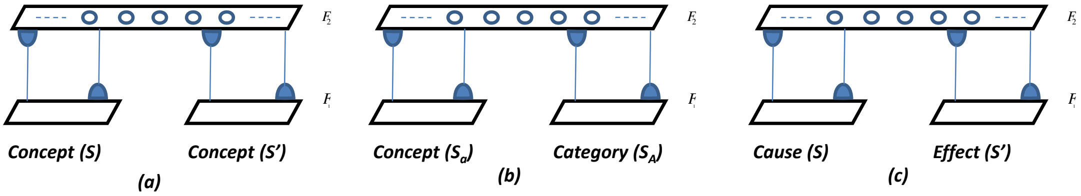


Figure 5.3: Different types of semantic knowledge and their neural network models: *(a)* association rule $(\mathbf{s}, \mathbf{s}')$ *(b)* concept hierarchy $\mathbf{s_a} : \mathbf{s_A}$ *(c)* causal relation $\mathbf{s} \rightarrow \mathbf{s}'$

shown, an attribute with a large discrepancy between its current value and expected (or primed) value will gain attention. For all attributes, the attention level gradually decays over time. The long-term memory modules perform memory retrieval by attending to the attributes with a high level of attention. This serves to confine the search within the newly discovered information.

5.5.2 Semantic Memory Model

In this architecture, the semantic memory is not a unitary system. In other words, there may be multiple types of semantic memory network in semantic memory, each represents a specific structure of knowledge. Each entry in semantic memory generalizes similar inputs into the same category rather than as separate entries. Each type of semantic memory can be made as a fusion ART with each input field representing a concept or memory state (s, s', s_a and s_A). The generalization can be achieved by lowering the vigilance parameter so that slightly different input patterns will still activate the same category. Figure 5.3 illustrates the various basic types of semantic memory.

5.5.3 Procedural Memory Model

The procedural memory model is based on a 3-channel Fusion ART model, namely Temporal Difference-Fusion Architecture for Learning, COgnition, and Navigation (i.e. TD-Falcon) [138], which learns the action and value policies through reinforcement learning across the sensory, motor, and feedback channels. The procedural memory model comprises a cognitive field F_2^c and three input fields, namely a sensory field F_1^{c1} for representing current states, an action field F_1^{c2} for representing actions, and a reward field F_1^{c3} for representing reinforcement values. Each F_2^c node represents a action rule in form of $P_k : s \rightarrow (a, r)$, which suggests a possible action a with a certain level of expected reward r (payoff), based on a given situation s.

Given the current state s, the model searches for optimal action a (with highest expected reward, r) through a direct code access procedure in the set of stored action rules. Upon receiving a feedback from the environment after performing the action, a TD (Temporal Difference) formula is used to compute a new estimate of the Q value of performing the chosen action in the current state. The new Q value is then used as the teaching signal for the procedural memory model to learn the association of the current state and the chosen action to the estimated Q value. The details are elaborated as described in [138].

5.6 An Illustrative Domain

The Toad and Frog puzzle [139] presents a standard reinforcement learning problem with a wide variety of semantic knowledge on the game domain and playing strategies. In the actual configuration, there is a seven-square board with the initially empty central square. Three toads and three frogs occupy the three leftmost and rightmost squares respectively (shown as Figure 5.4). The goal of the game is to swap the positions of the toads and frogs, such that the toads are placed in the three rightmost squares and the frogs are placed in the three leftmost squares. A square can be occupied by only an animal at a time in the game. Each type of animals can only move in one direction: toads can move only to the right and frogs only move to the left. In the game, an animal can only perform two types of feasible move: (1) a *Slide* if the next square in its moving direction is an empty square and (2) a *Jump* if the next square in its moving direction is the animal of the different type and the empty square is placed the two-square away from it in its moving direction. There are two symmetrical solution paths with the different type of animal first moved. Each solution has 15 moves, consisting of nine Jumps and six Slides.

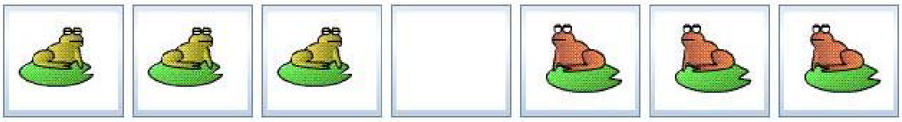


Figure 5.4: The initial configuration of a the Toad and Frog puzzle (image adopted from [8])

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). Besides the first move that allows two possible moves, there is a single correct move for every step in the right solution path.

5.6.1 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/empty) 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 5.1.

In the four 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

Table 5.1: Sample semantic rules on move validity

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

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

3) IF current_square is Frog, and square_right_1 is Space,
   THEN movable is true;

4) IF current_square is Frog, and square_right_1 is Toad,
      and square_right_2 is Space,
   THEN movable is true;
```

of the square n-square away on the left/right side of the current square. This type of semantic knowledge can be modelled as a fusion ART network as shown in Figure 5.5. The network model of semantic memory 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 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.

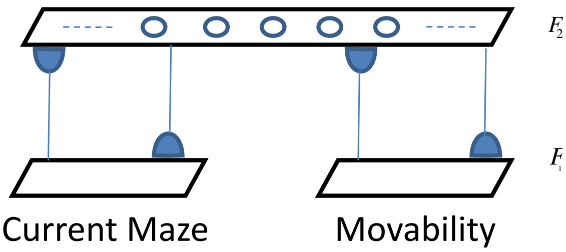


Figure 5.5: Semantic network model on move validity

Jump and Random Strategy: As derived by a previous cognitive study on the Toad

Table 5.2: Semantic rules to implement JRND strategy

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

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

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

4) IF MOVE1 is Slide, and MOVE2 is Slide,
   THEN PERFORM_MOVE1 is true, and PERFORM_MOVE2 is true;

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

and Frog puzzle [140], there exist several strategics which can improve the process of problem solving. Each of the available strategy 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. The 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 five semantic rules in Table 5.2.

In Rule 1 and 4, when both moves are valid, one of them will be selected randomly. Rule 5 indicates a forced move when only a single feasible move exists. The JRND strategy also requires additional knowledge on the feasible move(s) based on each possible puzzle state. This set of semantic knowledge uses two boolean post-conditional attributes to indicate the type of a move, as shown in Table 5.3.

Figure 5.6 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 5.6-a)

Table 5.3: Sample semantic rules on move type

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

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

3) IF current_square is Frog, and square_right_1 is Space,
   THEN is_slide is true;

4) IF current_square is Frog, and square_right_1 is Toad,
      and square_right_2 is Space,
   THEN is_jump is true;
```

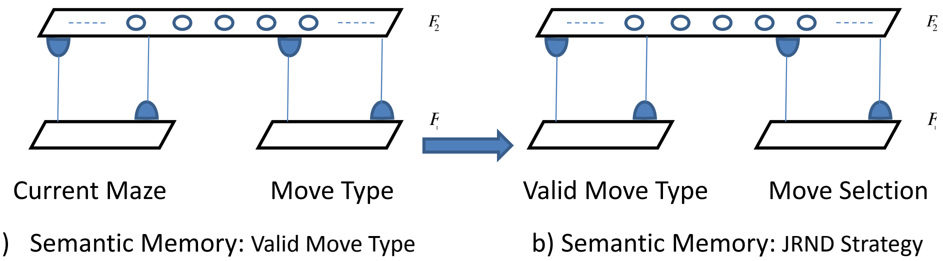


Figure 5.6: Semantic memory and interactions to implement JRND strategy: (1) activate network a) to determine feasible moves and their types based on the current puzzle status (2)the feasible moves and their types from a) is fed to network b) to recommend a single move

and a network to implement JRND (Figure 5.6-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.

5.6.2 Procedural Memory

The procedural memory learns and performs the move selection through the reinforcement learning algorithm as stated in Section 5.5.3. The state, action, and reward vectors in Figure 5.7 correspond to the input fields in a multi-channel ART network. Move pattern (i.e. moving one of the seven squares in the puzzle) in the state vector represents the move currently selected. The action vector indicates the next move to be selected. Based on the state field (stating the animal type currently occupied each of the seven squares in the puzzle) and the reward (set to the maximum), 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.

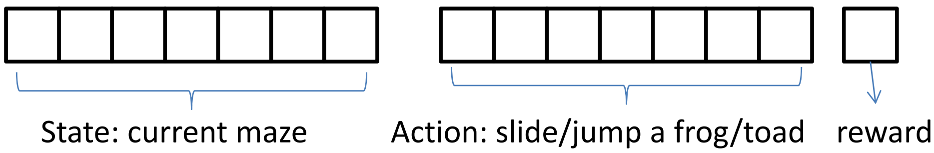


Figure 5.7: The input vectors to a procedural memory in Toad and Frog puzzle

The network learns by updating the weighted connections according to the feedback received and applying temporal difference methods as described by Section 5.5.3 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 leading 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.

5.6.3 Results and Discussion

Experiments are conducted to play the Toad and Frog 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 rate and the trial number of the first successful game, averaging over ten independent experiment runs.

	Success Rate (%)	First Successful Trial
Pure Procedural Learning (fusion ART based)	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	93.54	59.1

Table 5.4: Performance comparison with different memory options on solving the Toad and Frog puzzle

Table 5.4 shows the performance of five different experiment configuration embedded with different memory options, including (1) the pure procedural memory learning as described in Chapter 5.6.2; (2) the pure procedural memory learning using the standard Q-learning method based on the implementation presented in [141]; (3) the stand-alone semantic memory model with the knowledge on feasible moves; (4) the stand-alone semantic memory model implementing the Jump and Random (JRND) strategy; and (5) the procedural memory learning combined with the semantic knowledge on feasible moves. From Table 5.4, 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 procural) required by solving the Toad and Frog puzzle. The experiment also shows that the presented 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 combination with semantic memory, the experiment confirms that the interaction of the procedural and semantic memory modules further improves the learning resulting in a much better level of performance.

5.7 Empirical Study on StarCraft

In this section, we present the overall performance of the declarative-procedural memory systems by embedding the systems into the StarCraft game domain. StarCraft is a popular real-time strategy game developed by Blizzard Entertainment [142]. It presents the complete set of challenging domain tasks for both human players and the intelligent virtual agents. In each game, a player in StarCraft starts with a small territory and several working units. The player employs to continuously require its working units to harvest resources (mineral and gas) from its territory. The player is required to manage the resources collected in an effective manner to expand and constrict its territory, as well as increase its unit population. In order to win a game, the player eventually builds up its army force (i.e. a group of specific type of units and buildings) and destroy the opponents' territory. Figure 5.8 shows a screen snapshot from the game play in StarCraft.

Figure 5.8: A screen snapshot of StarCraft

The complicated tasks in StarCraft show a wide variety for challenges to artificial intelligence (AI) research, including imperfect information, multi-task planning, spatiotemporal reasoning and potential multi-agent communications. One great thing about StarCraft is that it offers an easy and standard way to build intelligent virtual agents in its game domain. This is accomplished by a software interface, namely BWAPI (An API for interacting with Starcraft:Broodwar) [143], to manage the interaction between a virtual agent and the game environments. With the help of BWAPI, some researchers have applied different types learning agents in StarCraft [144, 145, 146]. Furthermore, the success in this game requires the expertise level of skills and knowledge on the game domain, which also indicates a need of the multiple-memory systems for the complex knowledge acquisition and potential memory interactions.

5.7.1 Semantic Memory In StarCraft

Resource gathering, building construction and unit production are the three main tasks of 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 5.9. The proposed SM model has two input (F_1) fields: the Resource field represents the resource level in term of mineral, gas and supply count, while the Action field represents the various construction or production action.

Each node in F_2 represents a causal relation rule. Some examples of the causal rules can be described as Table 5.5.

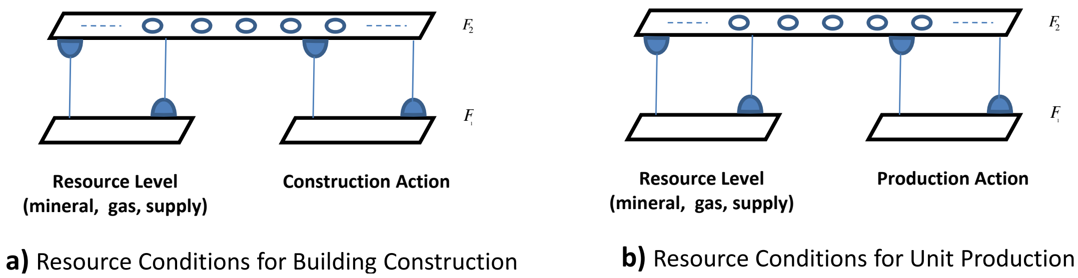


Figure 5.9: Representation of semantic knowledge on resource conditions to accomplish basic tasks in Starcraft

Table 5.5: Sample semantic rules on resource conditions to accomplish basic tasks in Starcraft

```
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


IF mineral is [400, +∞ )
   and gas is [0, +∞ )
  THEN HAVE_ENOUGH_RESOURCE_TO_BUILD_COMMAND_CENTER is true
```

5.7.2 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 PR with only the procedural memory, SP incorporating the multiple memory system using SP interaction, and PS using the PS interaction. The semantic knowledge employed includes the resource conditions for both building construction and unit production presented previously.

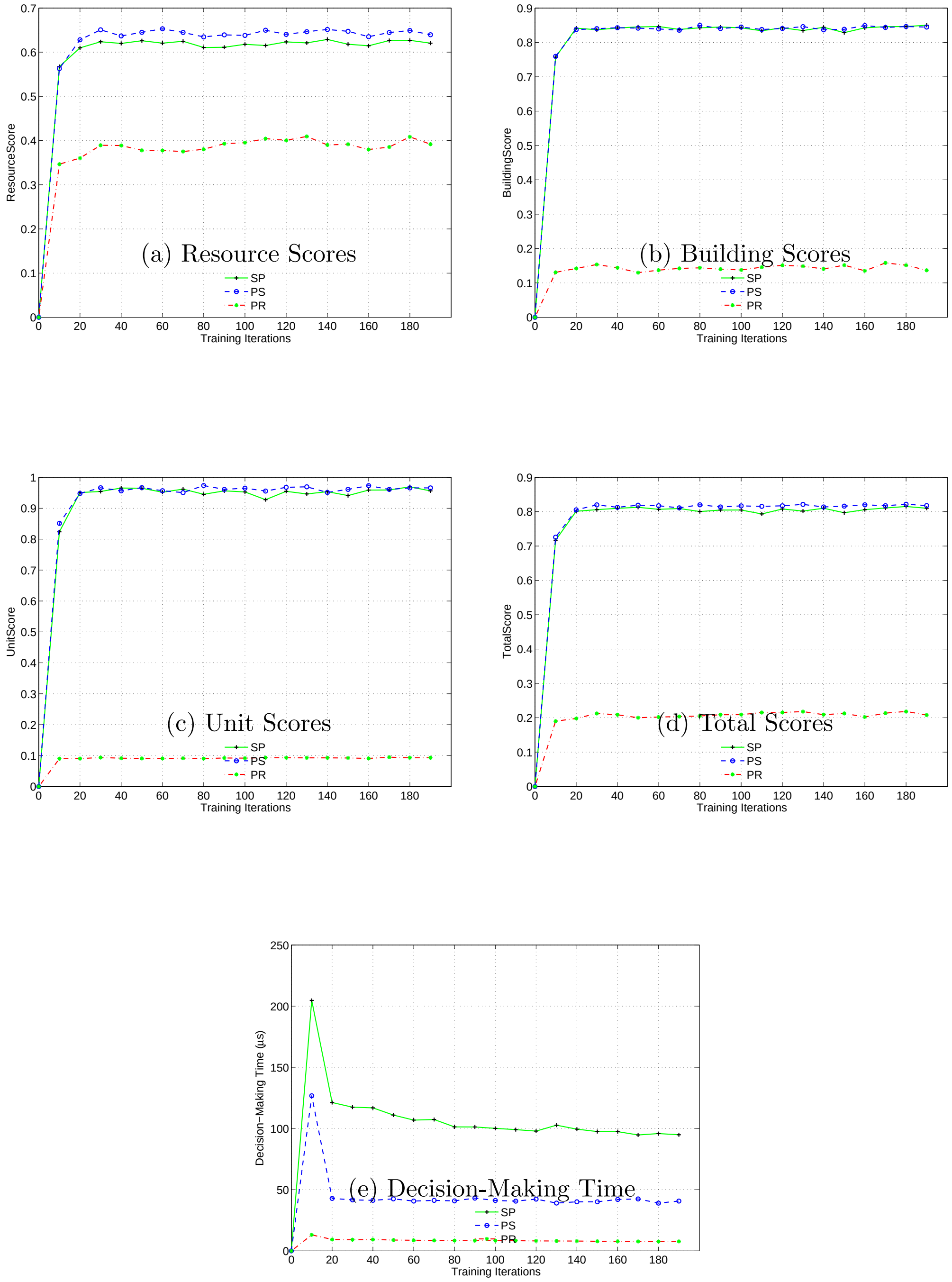


Figure 5.10: Performance comparison between learning agents using SP interaction, PS interaction and pure procedural learning (PR)

Experiments are conducted by letting PR, SP and PS to individually play in the StarCraft game. We compare the performance of different agents in terms of numbers of units, buildings and resources constructed or collected, at the end of each trial/game. Figure 5.10 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 weight sum of unit, building and resource count 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 5.10(a), SP and PS have provided a better performance in the task of resource management, comparing with that of PR. The use of semantic memory has helped the agents to collect the right amount of resource in terms of mineral, gas and supply level. PS has a slightly better performance compared with SP, 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 product of most of units requires some buildings as pre-condition. As shown in Figure 5.10(b), both the SP and PS interaction have helped to produce more units across various types, compared with that using pure procedural learning. While PR can only achieve an asset score of 0.3, both SP and PS have scored around 0.95 on these specific tasks. PS has a slightly better performance compared with SP, due to the same reason discussed.

From Figure 5.10(c), both SP and PS interaction provide a better performance on building construction, compared with pure procedural learning. As mentioned earlier, these tasks usually precede the task of unit production. More importantly, 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, SP and PS have led to faster building construction, facilitating the unit production task.

As shown in Figure 5.10(d) and (e), SP and PS interactions lead to a better overall performance during the game. PS produces a shorter decision time due to its targeted memory search. At the starting stage of the game, both SP and PS incur a longer decision time as procedural memory learning heavily depends on the retrievals from semantic memory. However, the decision time declines as the knowledge gradually transfers from semantic to procedural memory.

5.8 Summary

We have presented a multiple memory cognitive architecture with two major types of interaction between the semantic memory and the procedural memory. We illustrate how the two memory systems can cooperate in decision making process through the Toad and Frog puzzle and the Starcraft Broodwar game. Our experimental results have shown that the memory interactions consistently lead to a better performance in decision making.

Moving ahead, there are plans to explore other known forms of interaction between the semantic memory and the procedural memory. More complex learning tasks will also be defined to illustrate the use of the episodic memory and its interaction with the other memory systems.

Chapter 6

Conclusion

6.1 Summary of Contributions

This thesis presents a self-organizing approach to the explicit modeling of both procedural and declarative knowledge based on a unified set of computational principles and algorithms under fusion Adaptive Resonance Theory (fusion ART). More importantly, the multiple memory systems demonstrate how the various memory modules cooperate with and transfer knowledge from each other in order to achieve their individual roles and functionalities, as well as to facilitate the process of decision making. We present experimental studies, wherein the system is tasked to learn the procedural and declarative knowledge for the autonomous agents playing in various game domains. Our experimental results show that the multiple memory systems are able to enhance the performance of the agent in a real time environment by utilizing its both procedural and declarative knowledge. The main contributions of this dissertation work are summarized as follows.

1. We have presented a cognitive model, which consists of a declarative memory module, a procedural memory module, a working memory module and an intentional module. Each of the long-term memory modules in our system uses fusion ART as a building block to perform individual memory learning and support the different cognitive functionalities. we have also conducted an in-depth study into how

the interaction among declarative and procedural memory enables the cognitive model to produce a more versatile capability in decision making and problem solving. Specifically, we identify and formalize two major types of memory interaction and knowledge transfer processes between semantic memory and procedural memory. We have empirically investigated the performance of the overall declarative-procedural memory model using the Toad and Frog puzzle and an online strategic game known as Starcraft Broodwar. Our experimental results show that the model is able to learn procedural knowledge for the various tasks in both domains based on reinforcement learning signals from the environment. More importantly, the results also imply that the interaction between declarative knowledge and procedural skills can lead to a significant improvement on both learning efficiency and performance.

2. We have developed a declarative memory model consisting of episodic memory and semantic memory using fusion ART as building blocks. Episodic memory model stores specific experiences and their complex spatio-temporal relations. The contents in episodic memory can be selected to form the general facts and knowledge in semantic memory through a periodical knowledge transfer process.

3. Based on a generalization of fusion ART, we have shown how the neural network can be used in an episodic memory model, for encoding an individual's experience in the form of events as well as the spatio-temporal relations among events. The model supports complex-event storage through its multiple-channel learning capability inherited from fusion ART. An additional encoding scheme is also introduced, which allows complex sequences of events to be clustered and grouped. The model further incorporates a novel approximate memory search procedure, which performs parallel search of stored episodic traces continuously in response to potentially imperfect search cues. We have conducted empirical experimental evaluation

on the episodic memory model using a first-person shooting game, as well as a word recognition benchmark test. Various tests are performed on the built memory model to access its efficiency of possible memory retrieving during the games. The experimental results show that the model is able to provide a robust level of performance in encoding and recalling events and episodes even with incomplete and noisy cues. This is mainly due to its approximate retrieval using resonance search. The experiments conducted also indicate that forgetting promotes effective consolidation of its storage such that crucial knowledge can be kept in the memory while the size of the stored information is regulated by discarding trivial and noisy information.

4. We have proposed several semantic memory representations based on the same fusion ART encoding principle. Different from episodic memory, we view that the semantic memory is not unitary, in a way that different types of semantic memory network are required, each representing a different structure of knowledge. Through the interaction with the episodic memory module, a general procedure is also provided wherein the contents of episodic memory may be consolidated and transferred to a more permanent form of semantic memory. Our experiments on the integrated declarative memory model confirm that an explicit memory model can improve the agent learning and performance by acquiring useful knowledge for the task at hand through memory consolidation, relieving the agent from continuously reasoning and processing the information for learning. It is also demonstrated that the forgetting regulates the memory size while the performance is still being improved. Moreover, the experiment shows faster forgetting can result in better learning. This indicates that the forgetting can successfully filter out insignificant entries while maintaining the useful ones. The findings can inspire the exploration

of forgetting as a useful feature of intelligent agents and machine learning systems in general.

6.2 Future Work

There are many possible directions for further research on the modeling of multiple memory systems and learning. One key issue is the representation and learning of more general semantic structure. We shall also explore other roles and usages of episodic memory in decision making. More complex forms of interaction among the different memory models may also be identified and studied possibly in more advanced and complex domains. The rest of this section provide a detailed discussion of each potential future direction of this work.

6.2.1 Formation of Semantic Memory

The memory consolidation process requires the prior awareness on the specific types of semantic knowledge. The effective learning of semantic memory requires a fair judgement by developers on semantic knowledge to be learned. Therefore, the memory consolidation process may demand a deep understanding of the relevant tasks and domains. Additionally, the fixation on the types of learned knowledge also limits the functionalities of the model by preventing the discovery of novel knowledge types. To develop more useful and general purpose semantic memory, we may extend our multiple memory systems for automatically determination of which semantic knowledge are necessary and useful. One possible solution is to let episodic memory keep looking for significant relations among the stored information while learning, recalling and consolidation. The explored semantic knowledge to be learned shall be in the form of meaningful groupings among several event attributes and/or event patterns.

6.2.2 Roles of Episodic Memory

Our case study in the UT domain shows that the episodic memory model is able to enhance the agent performance by providing learning samples to form semantic knowledge on the weapon effectiveness. However, episodic memory model may also contribute directly to the performance of the agent rather than just as a transient structure. Take the UT game as an example, a more direct usage of episodic memory model is to let the agent predict the weapon effectiveness and choose the optimum weapon by retrieving and investigating prior weapon-firing experiences under similar situations stored in the episodic memory. In this way, the case-based learning by episodic memory can also be compared with the corresponding learning from semantic knowledge for their effectiveness to improve integrated performance. To investigate the performance of learning with episodic memory, the memory system model can be further applied to additional cognitive tasks wherein episodic memory has been identified with critical roles, for example navigation and goal processing.

6.2.3 Forms of Memory Interaction

Our overall architecture formalizes two types of interactions between semantic and procedural memories. However, additional forms of interactions between the three different long-term memory modules could be required to support more complex and meaningful cognitive tasks. Therefore, another potential extension of our work is to explore other forms of interactions, especially the interactions between the semantic memory and the procedural memory, as well as the ones between episodic memory and the procedural memory. One possible form of interaction between episodic and procedural memory is to retrieve specific past experience in episodic memory for use in decision making, as well as for driving the subsequent learning of new procedural memory. Moreover, more complex learning tasks can be identified to illustrate the use of more complex forms of memory interaction.

6.2.4 Attentional Control in Working Memory

Our working memory module incorporates an attentional mechanism which ensures that the information and knowledge important to the current task is always retained in the memory buffer. The importance of each piece of information held is evaluated mainly based on its recency and prediction surprises. However, a more realistic attentional mechanism should evaluate the information importance based on all related factors, such as rewards from the environment, relation to the current (sub)goals and emotions. The refined attentional mechanism should be evaluated using the additional domains, wherein the working memory, especially the attention control, has been identified with critical roles, for example navigation tasks.

6.2.5 Learning in Complex Domains

In this research, we have evaluated the performance of the multiple memory systems through (1) a first person shooting game and (2) a strategic game. However, the multiple memory systems can be applied to more complex cognitive functionalities and tasks. One possible cognitive capability supported by the memory model is the management of long term goals, wherein the episodic memory informs the agent whether or not its goals have been achieved. Another possible example can be prediction — the agent can retrieve a relevant episode to predict what comes next, how its actions affect the world, and the chance of success/failure. In order to investigate the performance and robustness of the multiple memory model while achieving higher level cognitive functionalities, we may need to move to more advanced domains for complex goals and scenarios.

List of Publications

Conference Papers:

- W. Wang, B. Subagdja, and A.-H. Tan. **A Self-Organizing Approach to Episodic Memory Modeling.** *In Proceedings, International Joint Conference on Neural Network, IEEE World Congress on Computational Intelligence*, pages 447-454, Barcelona, Spain, July 18-23, 2010.

- W. Wang, B. Subagdja, A.-H. Tan and Y.-S. Tan. **A Self-Organizing Multi-Memory System for Autonomous Agents.** *In Proceedings, International Joint Conference on Neural Networks, IEEE World Congress on Computational Intelligence*, pp. 480-487, Brisbane, Australia, June 10-15, 2012.

- B. Subagdja, W. Wang, A.-H. Tan, Y.-S. Tan, and L.-N. Teow. **Memory Formation, Consolidation, and Forgetting in Learning Agents.** *In Proceedings, Eleventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, pp. 1007-1014, Valencia, Spain, June 4-8, 2012.

- W. Wang, A.-H. Tan, Y.-S. Tan, and L.-N. Teow. **Declarative-Procedural Memory Interaction in Learning Agents.** *In Proceedings, 13th International Conference on Autonomous Agents and Multiagent Systems(AAMAS2014)*, Paris, France, May 5-9, 2014.

Journal Papers:

- W. Wang, B. Subagdja, A.-H. Tan and J. A. Starzyk. **Neural Modeling of Episodic Memory: Encoding, Retrieval, and Forgetting.** *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 23, No. 10 (October 2012), pp. 1574-1586.
- W. Wang and A.-H. Tan. **Neural Modeling of Semantic Memory In Learning Agents.** In submission to *IEEE Transactions on Systems, Man, and Cybernetics.*

Table of Acronym

ART Adaptive Resonance Theory

ASL Australian Sign Language

NPC Non-Player-Character

SMRITI System for the Memorization of Relational Instances from Temporal Impulses

CLS Complementary Learning Systems

TESMECOR Temporal Episodic and Semantic Memory using Combinatorial Representations

TCM Temporal Context Model

POMDP Partially Observable Markov Decision Process

FCM Fuzzy Cognitive Maps

ACT-R Adaptive Control of ThoughtRational

LSA Latent Semantic Analysis

REM-II Retrieving Effectively from Memory-II

SRN Simple Recurrent Network

TD-FALCON Temporal Difference-Fusion Architecture for Learning, Cognition, and Navigation

CLARION Connectionist Learning with Adaptive Rule Induction On-line

SMSC Standard Model of System Consolidation

MTT Multiple Trace Theory

CTT Competitive Trace Theory

iFALCON intentional Fusion Architecture for Learning, COgnition and Navigation

EM Episodic Memory

HMM hidden Markov model

UT Unreal Tournament

SM Semantic Memory

References

[1] A. Nuxoll and J. E. Laird. Extending cognitive architecture with episodic memory. In *AAAI*, pages 1560–1564. AAAI Press, 2007.

[2] L. Shastri. Episodic memory and cortico-hippocampal interactions. *TRENDS in Cognitive Sciences*, 6(4):162–168, April 2002.

[3] 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.

[4] J. L. Elman. Finding structure in time. volume 14, pages 179–211, 1990.

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

[6] R. Sun, P. Slusarz, and C. Terry. The interaction of the implicit and the explicit in skill learing: A dual-process approach. *Psycological Review*, 112:159–192, 2002.

[7] R. Sun and X. Zhang. Top-down versus bottom-up learning in skill acquisition. In *Proceedings of the 24th Annual Conference of the Cognitive Science Society*, 2002.

[8] The frog puzzle: A solution. Web: http://britton.disted.camosun.bc.ca/frog_puzzle _sol.htm.

[9] E. Tulving. How many memory system are there? *American Psychologist*, 40:385–398, 1985.

[10] E. Kandel, J. Schwartz, and T. Jessell, editors. *Principles of Neural Science*. McGraw-Hill, 4^{th} edition, 2000.

[11] E. Tulving. Episodic and semantic memory. In Endel Tulving and W. Donaldson, editors, *Organization of Memory*, pages 381–403. Academic Press, New York, 1972.

[12] E. Tulving, editor. *Elements of Episodic Memory*. Oxford University Press, 1^{st} edition, 1983.

[13] M. A. Conway. *Exploring episodic memory*, volume 18, chapter 1.2, pages 19–29. Elsevier, 2008.

[14] S. Zola-Morgan and L. R. Squire. Neuroanatomy of memory. *Annual Review of Neuroscience*, 16:547–563, 1993.

[15] D. M. McBride. Methods for measuring conscious and automatic memory: A brief review. *Journal of Consciousness Studies*, 14(1):198–215, 2007.

[16] R. Gupta, C. M. Duff, N. L. Denburg, N. J. Cohen, A. Bechara, and D. Tranel. Declarative memory is critical for sustained advantageous complex decision-making. *Neuropsychologia*, 47:1686–1693, 2009.

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

[18] E. Tulving. Precis of elements of episodic memory. In *The Behavioral and Brain Sciences*, volume 7, pages 223–268. Cambridge University Press, 1984.

[19] 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.

[20] A. B. Samsonovvich and G. A. Ascoli. A simple neural network model of the hippocampus suggesting its pathfinding role in episodic memory retrieval. *Learning and Memory*, 12:193–208, 2005.

[21] The free encyclopedia. Web: http://www.wikipedia.org/.

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

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

[24] K. Lund, C. Burgess, and R. A. Atchley. Semantic and associative priming in high-dimensional semantic space. In J.D. Moore and J.F. Lehman, editors, *Proceedings of the 17th annual meeting of the Cognitive Science Society*, pages 660–665, Pittsburgh, 1995. Erlbaum.

[25] T. K. Landauer and S. T. Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240, 1997.

[26] 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.

[27] 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.

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

[29] G. A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63:81–97, 1956.

[30] D. C. Berry and Z. Dienes. *Implicit Learning: Theoretical and Empirical Issues.* UK: Lawrence Erlbaum Associates, 1993.

[31] A. Cleeremans and J. L. McClelland. Learning the structure of event sequences. *Journal of Experimential Psychology: General*, 120(3):235–253, 1991.

[32] P. Perruchet. and A. Vinter. PARSER: A model for word segmentation. *Journal of Memory and Language*, 39:246–263, 1998.

[33] R. Sun. *Duality of the Mind: A Bottom-Up Approach Toward Cognition.* Lawrence Erlbaum, Mahwah, 2002.

[34] 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, LNCS*, volume 4491, pages 1098–1107, Nanjing, China, June 2007.

[35] G. E. Rawlinson. The significance of letter position in word recognition. *Ph.d. dissertation*, 1976.

[36] M. W. Kadous. *Temporal classification: Extending the classification paradigm to multivariate analysis.* PhD thesis, School of Computer Science and Engineering, University of New South Wales, 2002.

[37] D. Wang, B. Subagdja, and A.-H. Tan. Creating human-like autonomous players in real-time first person shooter computer games. In *Proceedings, Twenty-First Annual Conference on Innovative Applications of Artificial Intelligence*, 2009.

[38] J. M. Gardiner. Episodic memory and autonoetic consciousness: a first-person approach. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 356:1351–1361, 2001.

[39] E. Tulving. *Where in the brain is the awareness of one's past?*, chapter 7, pages 208–228. Harvard Unversity Press, cambridge, 2000.

[40] W. C. Ho, K. Dautenhahn, and C. L. Nehaniv. Comparing different control architectures for autobiographic agents in static virtual environments. In Thomas Rist, Ruth Aylett, Daniel Ballin, and Jeff Rickel, editors, *Intelligent Virtual Agents, Fourth International Workshop, Kloster Irsee, Germany, 2003*, volume 2792 of *Lecture Notes in Computer Science*, pages 182–191. Springer, 2003.

[41] S. A. Vere and T. W. Bickmore. A basic agent. *Computational Intelligence*, 6:41–60, 1990.

[42] R. M. Shiffrin and M. Steyvers. A model for recognition memory: Rem – retrieving effectively from memory. *Psychonomic Bulletin & Review*, 4:145–166, 1997.

[43] A. B. Nelson and R. M. Shiffrin. Sarkae – modeling the co-evolution of event memory and knowledge. In *Proceedings of the 32nd Annual Conference of the Cognitive Science Society*, pages 254–259, 2010.

[44] R. M. Jones, J. E. Laird, P. E. Nielsen, K. J. Coulter, P. Kenny, and F. V. Koss. Automated intelligent pilots for combat might simulation. *AI Magazine*, 20:27–41, 1999.

[45] R. E. Wray, J. E. Laird, A. Nuxoll, D. Stokes, and A. Kerfoot. Synthetic adversaries for urban combat training. *AI Magazine*, 26:82–92, 2005.

[46] A. Nuxoll. *Enhancing Intelligent Agents with Episodic Memory.* PhD thesis, University of Michigan, 2007.

[47] G. Grossberg, W. John, and L. Merrill. The hippocampus and cerebellum in adaptively timed learning, recognition, and movement. *Cognitive Neuroscience*, 8:257–277, 1996.

[48] H. Masun. A hybrid neural network for arabic internet navigator commands recognition. In *Proceedings of the 2004 International Conference on Information and Communication Technologies: From Theory to Applications*, pages 607–608, April 2004.

[49] C. J. Wu. Dart1 — a possible ideal building block of memory systems. In *Proceedings of the 1994 IEEE World Congress on Computational Intelligence*, volume 2, pages 1086–1091, June 1994.

[50] S. E. Taylor, M. J. Healy, T. P. Caudell, N. J. Cohen, P. Watson, S. J. Verzi, J. D. Morrow, M. L. Bernard, and H. Eichenbaum. Memory in silico: building a neuromimetic episodic cognitive model. In *Proceedings of 2009 WRI World Congress on Computer Science and Information Engineering*, volume 5, pages 733–737, April 2009.

[51] 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.

[52] R. C. O'Reilly and J.W. Rudy. Conjunctive representations in learning and memory: principles of cortical and hippocampal function. *Psychological Review*, 108:311–345, 2001.

[53] K. A. Norman and C. Oeilly. Modeling hippocampal and neocortical contributions to recognition memory: A complementary learning systems approach. *Psychological Review*, 110:611–646, 2003.

[54] R. C. O'Reilly, R. Bhattacharyya, M. D. Howard, and N. Ketza. Complementary learning systems. *Cognitive Science*, 20, 2011.

[55] R. C. O'Reilly, K. A. Norman, and J. L. McClelland. A hippocampal model of recognition memory. In *NIPS*, 1997.

[56] J. A. Starzyk and H. He. Anticipation-based temporal sequences learning in hierarchical structure. *IEEE Transactions on Neural Networks*, 18(2):344–358, 2007.

[57] J. A. Starzyk and H. He. Spatio-temporal memories for machine learning: A long-term memory organization. *IEEE Transactions on Neural Networks*, 20(5):768–780, 2009.

[58] V. A. Nguyen, J. A. Starzyk, A. Tay, and W.-B. Goh. Spatio-temporal sequence learning of visual place cells in robotic navigation. In Thomas Rist, Ruth Aylett, Daniel Ballin, and Jeff Rickel, editors, *International Joint Conference on Neural Network, IEEE World Congress on Computational Intelligence*, 2010.

[59] V. A. Nguyen, J. A. Starzyk, W-B. Goh, and D. Jachyra. Neural network structure for spatio-temporal long-term memory. *IEEE Transactions on Neural Networks and Learning Systems*, 23:866–875, 2012.

[60] V. A. Nguyen, J. A. Starzyk, and W-B. Goh. A spatio-temporal long-term memory approach for visual place recognition in mobile robotic navigation. *Robotics and Autonomous Systems*, 61:1744–1758, 2013.

[61] D. Wang and M. A. Arbib. Timing and chunking in processing temporal order. *IEEE Transactions on Systems, Man, And Cybernetics*, 23(4):993–1009, 1993.

[62] G. Mensink and J. G. Raaijmakers. a model for interference and forgetting. *Psychological Review*, 95:434–455, 1988.

[63] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.

[64] E. A. Zilli and M. E. Hasselmo. The influence of markov decision process structure on the possible strategic use of working memory and episodic memory. *PLoS ONE*, 3(7):e2756, 2008.

[65] E. A. Zilli and M. E. Hasselmo. Modeling the role of working memory and episodic memory in behavioral tasks. *Hippocampus*, 18:193–209, 2008.

[66] E. J. Sondik. *The optimal control of partially observable Markov processes*. PhD thesis, Stanford University, 1971.

[67] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.

[68] E. Hansen. Solving pomdps by searching in policy space. In *Proceedings of the Fourteenth International Conference on Uncertainty In Artificial Intelligence*, 1998.

[69] F. Leconte, F. Ferland, and F. Michaud. Fusion adaptive resonance theory networks used as episodic memory for an autonomous robot. In *Artificial General Intelligence*, volume 8598 of *Lecture Notes in Computer Science*, pages 63–72. Springer, 2014.

[70] R.C. O'Reilly and M.J. Frank. Making working memory work: A computational model of learning in the frontal cortex and basal ganglia. *Neural Computation*, 18:283–328, 2006.

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

[72] D. R. Hofstadter and M Mitchell. *Fluid Concepts and Creative Analogies*, chapter The copycat project: A model of mental fluidity and analogy-making, pages 205–267. Basic Books, 1995.

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

[74] L. Shastri and V. Ajjanagadde. From simple associations to systematic reasoning. *Behavioral and Brain Sciences*, 16:417–494, 1993.

[75] Y. Wang and J. E. Laird. Integrating semantic memory into a cognitive architecture. Tech. Rep. CCA-TR-2006-02, Center for Cognitive Architectures, University of Michigan, June 2007.

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

[77] Z. Dienes. Connectionist and memory-array models of artificial grammar learning. volume 16, pages 41–79, 1992.

[78] Z. Dienes, G. Altmann, and S.-J. Gao. Mapping across domains without feedback: A neural network model of transfer of implicit knowledge. volume 23, pages 53–82, 1999.

[79] A. Cleeremans. Mechanisms of implicit learning: Connectionist models of sequence processing. 1993.

[80] A.-H. Tan. Direct code access in self-organizing neural architectures for reinforcement learning. In *International Joint Conference on Artificial Intelligence*, pages 1071–1076, 2007.

[81] A. S. Reber and S. Lewis. Implicit learning: An analysis of the form and structure of a body of tacit knowledge. volume 114, pages 14–24, 1977.

[82] E. Servan-Schreiber and J. R. Anderson. Learning artificial grammar with competitive chunking. volume 16, pages 592–608, 1990.

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

[84] J. R. Anderson. Spanning seven orders of magnitude: a challenge for cognitive modeling. *Cognitive Science*, 26:85–112, 2002.

[85] J. E. Laird, P. S. Rosenbloom, and A. Newell. Chunking in soar: The anatomy of a general learning mechanism. *Machine Learning*, 1:11–46, 1986.

[86] S. Nason and J. E. Laird. Soar-rl, integrating reinforcement learning with soar. In *International Conference on Cognitive Modeling*, 2004.

[87] P. Slusarz and R. Sun. The interaction of explicit and implicit learning: An integrated model. In *Proceedings of Cognitive Science Society Conference*, pages 952–957, 2001.

[88] L. Nadel and M. Moscovitch. Memory consolidation, retrograde amnesia and the hippocampal complex. *Current Opinion in Neurobiology*, 7(2):217–227, 1997.

[89] L. R. Squire and P. Alvarez. Retrograde amnesia and memory consolidation: A neurobiological perspective. *Current Opinion in Neurobiology*, 5:169–177, 1995.

[90] G. Winocura, M. Moscovitchd, and B. Bontempif. Memory formation and long-term retention in humans and animals: convergence towards a transformation account of hippocampal-neocortical interactions. *Neuropsychologia*, 48(8):2339–2356, 2010.

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

[92] M. T. Ullman. chapter The declarative/procedural model of language. Sage Publications.

[93] M. P. Walker, T. Brakefield, J. A. Hobson, and R. Stickgold. Dissociable stages of human memory consolidation and reconsolidation. *Nature*, 425:616–620, 2003.

[94] R. M. Brown and E. M. Robertson. Off-line processing: reciprocal interactions between declarative and drocedural memories. *The Journal of Neuroscience*, 27:10468–10475, 2007.

[95] R. Sun, X. Zhang, and R. Mathews. Modeling meta-cognition in a cognitive architecture. In *Proceedings of the 27th Annual Conference of the Cognitive Science Society*, 2005.

[96] W. Wang, B. Subagdja, and A.-H. Tan. A self-organizing approach to episodic memory modeling. In *Proceedings, International Joint Conference on Neural Network, IEEE World Congress on Computational Intelligence*, pages 447–454, 2010.

[97] W. Wang, B. Subagdja, A.-H. Tan, and J. A. Starzyk. Neural modeling of episodic memory: Encoding, retrieval, and forgetting. *IEEE Transactions on Neural Networks*, submitted.

[98] E. Dere, A. Easton, and L. Nadel, editors. *Handbook of Episodic Memory*, volume 18. Elsevier, Amsterdam, 2008.

[99] L. Shastri. Biological grounding of recruitment learning and vicinal algorithms in long-term potentiation. In *Emergent Neural Computational Architectures Based on Neuroscience*, pages 348–367, 2001.

[100] 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.

[101] S. Grossberg. Behavioral contrast in short-term memory: Serial binary memory models or parallel continuous memory models? *Journal of Mathematical Psychology*, 3:199–219, 1978.

[102] G. Bradski, G. A. Carpenter, and S. Grossberg. STORE working memory networks for storage and recall of arbitrary temporal sequences. *Biological Cybernetics*, 71:469–480, 1994.

[103] B. Subagdja and A.-H. Tan. A self-organizing neural network architecture for intentional planning agents. In *Eighth International Conference on Autonomous Agents and Multiagent Systems*, pages 1081–1088, 2009.

[104] F. I. Bashir, A. A. Khokhar, and D. Schonfeld. Object trajectory-based activity classification and recognition using hidden markov models. *IEEE Transactions on Image Processing*, 18:1912–1919, 2007.

[105] A. Naftel and S. Khalid. Classifying spatiotemporal object trajectories using unsupervised learning in the coefficient feature space. *Multimedia System*, 12:227–238, 2006.

[106] S. Grossberg and J. W. Merrill. The hippocampus and cerebellum in adaptively timed learning, recognition, and movement. *Cognitive Neuroscience,*, 8:257–277, 1996.

[107] L. Shastri. Recruitment of binding and binding-error detector circuits via long-term potentiation. *Neurocomputing*, 27:865–874, 1999.

[108] G. J. Rinkus. Context-sensitive spatio-temporal memory. In *Proceedings of the 1993 World Congress on Neural Networks*, pages 344–347, 1993.

[109] G. J. Rinkus. Temecor: an associative, spatiotemporal pattern memory for complex state sequences. In *Proceedings of the 1995 World Congress on Neural Networks*, pages 442–448, 1995.

[110] W. Wang, B. Subagdja, and A.-H. Tan. A self-organizing neural model for episodic memory formation and consolidation. Technical report, SCE.NTU, 2011.

[111] W. C. Ho, K. Dautenhahn, and C. L. Nehaniv. Computational memory architectures for autobiographic agents interacting in a complex virtual environment: A working model. *Connection Science*, 20(1):21–65, 2008.

[112] C. Brom, T. Korenko, and J. Lukavsk. How do place and objects combine? what-where? memory for human-like agents. In Zsfia Ruttkay, Michael Kipp, Anton Nijholt, and Hannes Vilhjlmsson, editors, *Intelligent Virtual Agents*, volume 5773 of *Lecture Notes in Computer Science*, pages 42–48. Springer Berlin / Heidelberg, 2009.

[113] T. Ribot. *Diseases of the Memory: An Essay in the Positive Psychology.* D. Appleton and Company, New York, NY, 1882.

[114] B. J. Knowlton, L. R. Squire, and R. E. Clark. Retrograde amnesia. *Hippocampus*, 11:50–55, 2001.

[115] M. Meeter and J. M. Murre. Consolidation of long-term memory: Evidence and alternatives. *Psychological Bulletin*, 130:843–857, 2004.

[116] L. R. Squire, Cohen N. H., and L. Nadel. *The medial temporal region and memory consolidation: a new hypothesis*, chapter Memory consolidation, pages 185–210. Hillsdale, NJ: Erlbaum and Associates, 1984.

[117] L. R. Squire. Memory and the hippocampus: a synthesis from findings with rats, monkeys, and humans. *Psychological Review*, 99:195–231, 1992.

[118] H. L. Roediger, Y. Dudai, and S. M. Fitzpatrick. *Science of memory: concepts.* New York, NY: Oxford University Press, 2007.

[119] P. W. Frankland and B. Bontempi. The organization of recent and remote memories. *Nature Reviews Neuroscience*, 6:119–130, 2005.

[120] Y. Dudai. The neurobiology of consolidations, or, how stable is the engram? *Annual Review of Psychology*, 55:51–86, 2004.

[121] L. Shastri and S. Chang. A spatiotemporal connectionist model of algebraic rule-learning. Technical Report TR-99-011, ICSI, Berkeley, 1999.

[122] Unreal tournament. Web: http://http://www.unrealtournament.com/blog/.

[123] Pogamut: virtual characters made easy, tutorials, pogamut devel wiki, latest pogamut lectures. Web: http://pogamut.cuni.cz/main/tiki-index.php.

[124] D. Gamez, Z. Fountas, and A. K. Fidjeland. A neurally controlled computer game avatar with humanlike behavior. *IEEE Transactions on Computational Intelligence and AI in Games*, 5:1–14, 2013.

[125] F. G. Glavin and M. G. Madden. Dre-bot: a hierarchical first person shooter bot using multiple sarsa(α) reinforcement learners. In *Proceedings of the 2012 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 148–152, 2012.

[126] M. A. Wood and J. J. Bryson. Skill acquisition through program-level imitation in a real-time domain. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 37:272–285, 2007.

[127] C. Rosenthal and C. B. Congdon. Personality profiles for generating believable bot behaviors. In *Proceedings of the 2012 International Conference onComputer Games (CGAMES)*, pages 124–131, 2012.

[128] A. Berler and S. E. Shimony. Multi-agent handling of opportunism: Awol meets discretized 'unreal tournament'. In *Proceedings of the 2006 IEEE International Conference on Systems, Man and Cybernetics*, pages 2505–2510, 2006.

[129] H. Wang, Y. Gao, and X. Chen. Rl-dot: a reinforcement learning npc team for playing domination games. *IEEE Transactions on Computational Intelligence and AI in Games*, 3:17–26, 2010.

[130] S. Megan, L.-U. Stephen, and M. Héctor. Retaliate: learning winning policies in first-person shooter games. In *Proceedings of the 2007 National Conference on Innovative Applications of Artificial Intelligence*, pages 1801–1806. AAAI Press, 2007.

[131] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, 1998.

[132] W. Wang, B Subagdja, A.-H Tan, and Y.-S. Tan. A self-organizing multi-memory system for autonomous agents. In *Proceedings, 2012 International Joint Conference on Neural Networks*, pages 252–258, 2012.

[133] B Subagdja, W. Wang, , A.-H Tan, Y.-S. Tan, and L.-N. Teow. Memory formation, consolidation, and forgetting in learning agents. In *In Proceedings, Eleventh International Conference on Autonomous Agents and Multiagent Systems*, pages 1007–1014, 2012.

[134] R. A. Poldrack and J. D. Gabrieli. Characterizing the neural mechanisms of skill learning and repetition priming: Evidence from mirror reading. *Brain*, 124(1):67–82, 2001.

[135] O. Baumann, E. Chan, and J. B. Mattingley. Hippocampal, parahippocampal and striatal neuronal activity predicts object-location retrieval during active navigation. In *Proceedings of the 9th Conference of the Australasian Society for Cognitive Science*.

[136] N. Cowan. What are the differences between long-term, short-term, and working memory? In *Progress in Brain Research*, volume 169, page 323?38. Elsevier, 2008.

[137] T. Shallice and E. K. Warrington. Independent functioning of verbal memory stores: a neuropsychological study. *Quarterly Journal of Experimental Psychology*, 22:261–273, 1970.

[138] 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.

[139] M. Cary and R. A. Carlson. Distributing working memory resources during problem solving. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 27:836–848, 2001.

[140] 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.

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

[142] Starcraft: Brood war. http://starcraft.wikia.com/wiki/StarCraft:_Brood_War.

[143] Bwapi: An api for interacting with starcraft: Broodwar. Web: https://code.google.com/p/bwapi/.

[144] S. Wender and I. Watson. Applying reinforcement learning to small scale combat in the real-time strategy game starcraft:broodwar. In *Proceedings of the 2012 IEEE Conference on Computational Intelligence and Games (CIG)*, 2012.

[145] A. Micic, D. Arnarsson, and V. Jonsson. Developing game ai for the real-time strategy game starcraft. Technical report, Reykjavik University, 2011.

[146] A. Shantia, E. Begue, and M. Wiering. Connectionist reinforcement learning for intelligent unit micro management in starcraft. In *Proceedings of the 2011 International Joint Conference on Neural Networks (IJCNN)*, pages 1794–1801, 2011.