Spring 1-1-2013

The importance of medial prefrontal cortex in resolving interference during memory retrieval

Srinimisha Morkonda Gnanasekaran
University of Colorado at Boulder, srinimisha@colorado.edu

Follow this and additional works at: https://scholar.colorado.edu/csci_gradetds
Part of the Neurosciences Commons

Recommended Citation
Morkonda Gnanasekaran, Srinimisha, "The importance of medial prefrontal cortex in resolving interference during memory retrieval" (2013). Computer Science Graduate Theses & Dissertations. 68.
https://scholar.colorado.edu/csci_gradetds/68

This Thesis is brought to you for free and open access by Computer Science at CU Scholar. It has been accepted for inclusion in Computer Science Graduate Theses & Dissertations by an authorized administrator of CU Scholar. For more information, please contact cuscholaradmin@colorado.edu.
The importance of medial prefrontal cortex in resolving interference during memory retrieval

by

Srinimisha Morkonda Gnanasekaran

B.S., University of Reading, 2011

A thesis submitted to the
Faculty of the Graduate School of the
University of Colorado in partial fulfillment
of the requirements for the degree of
Master of Science
Department of Computer Science
2013
This thesis entitled:
The importance of medial prefrontal cortex in resolving interference during memory retrieval
written by Srinimisha Morkonda Gnanasekaran
has been approved for the Department of Computer Science

Prof. Randall C. O’Reilly

Prof. Michael Mozer

Prof. James Martin

Date ________________

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.
The prefrontal cortex (PFC) is considered as the brain’s executive, with multiple functions and this executive role has been studied for a long time. The role of PFC, especially the medial prefrontal cortex (mPFC), also extends to resolving conflicts during a memory retrieval task. This has also been confirmed in imaging studies that show activations of PFC during a memory retrieval task. A recent study with rats has shown the significance of mPFC in memory retrieval task that involves selecting the right response from competing targets, which is, resolving the conflict among multiple similar choices to make a final decision. To demonstrate the conflict-resolving role of mPFC in a computational paradigm, we put forth a neural network based computational model of the hippocampus and the mPFC. The motivation is to build a model that represents the behavioral results from the experiment conducted by Peters et al., (2013). Our simulations compare the performance of the model when the mPFC is present and absent. Our results show that when the mPFC is absent, the ability to learn multiple conflicting items is impaired. We also investigate to see if the memories that are learnt in absence of mPFC pose any effects on acquisition of new conflicting memories. Results show that the memories encoded without the mPFC did not pose any interference to any new memories that are to be learnt with mPFC. Our results on the whole suggested that mPFC has a critical role in resolving interference in memory encoding and retrieval.
Dedication

To my family.
Acknowledgements

I would like to thank all the people who contributed to this thesis work. I like to thank my advisor, Dr. Randall C. O’Reilly who has been very supportive by providing his guidance all along the project. His time, effort and support have been detrimental to this thesis work. I thank the committee, Dr. Michael Mozer and Dr. James Martin for their feedback and suggestions on the work. I would also like to thank Greg Peters and his colleagues for their work, which was the motivation for this thesis. My hearty thanks to my lab mates who have been providing feedback and comments that made improvements in the work. Finally, I thank my family and friends who constantly provide moral support throughout my life.
## Contents

1 Introduction 1

2 Computational Model and its learning framework 6
   2.1 Leabra Learning Framework 6
   2.2 Hippocampus Model 9
   2.3 mPFC 13

3 The role of mPFC in acquisition of multiple conflicting items 15
   3.1 Simulation Method 16
   3.2 Results and Discussion 19

4 The contributions of mPFC during encoding 24
   4.1 Simulation Method 24
   4.2 Results and Discussion 25
   4.2.1 Observations under a modified environment 29

5 Discussion 32
   5.1 Context representation in mPFC vs. Hippocampus 34
   5.2 Future Work 35
   5.3 Conclusions 35
Tables

Table

3.1 The details of the two cases and their training environments under which the simulation 1 were performed. .................................................. 18

3.2 Table shows the % correct of List 2 at the end of training of 6 epochs (in our simulation) and 3 days (in behavioral experiment). .............................. 20

3.3 Table shows the % correct of List 2 at the end of training of 10 epochs (in our simulation) and 5 days (in behavioral experiment). ....................... 21

4.1 The details of the two cases and their training environments under which the simulation 2 were performed. .................................................. 25

4.2 Table summarizing the results from our simulation and behavioral experiment. It shows the % correct of List 2 at the end of training of 10 epochs (in our simulation) and 5 days (in behavioral experiment). ............................. 28

4.3 Table summarizing the results from our modified simulation environment. It shows the % correct of List 2 at the end of training of 10 epochs. ....................... 29
## Figures

**Figure**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Leabra learning mechanisms. Figure adopted from [1]</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>Computational model as used for the simulations</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>Schematic diagram of the Computational Model</td>
<td>12</td>
</tr>
<tr>
<td>3.1</td>
<td>Input format of the Computational Model</td>
<td>18</td>
</tr>
<tr>
<td>3.2</td>
<td>Simulation results of the acquisition of list 2 after learning list 1 either in absence or presence of mPFC environment</td>
<td>20</td>
</tr>
<tr>
<td>3.3</td>
<td>Behavioral results of the acquisition of list 2 after learning list 1 either in absence or presence of mPFC environment. Figure adopted from [2]</td>
<td>21</td>
</tr>
<tr>
<td>3.4</td>
<td>Interference Index reflecting decline in performance of learning list 2 after learning list 1</td>
<td>22</td>
</tr>
<tr>
<td>4.1</td>
<td>Performance of list 2 where list 1 is learnt in absence of mPFC and list 2 is learnt in presence of mPFC</td>
<td>26</td>
</tr>
<tr>
<td>4.2</td>
<td>Acquisition of list 2 after learning of list 1 either in presence of absence of mPFC. Figure adopted from [2]</td>
<td>27</td>
</tr>
<tr>
<td>4.3</td>
<td>Performance of list 2 where list 1 is learnt in absence of mPFC and list 2 (with fully repeated items from list 1) is learnt in presence of mPFC</td>
<td>30</td>
</tr>
</tbody>
</table>
The prefrontal cortex (PFC) is one of the vital regions of the brain that has multiple functions, especially in humans. Having a very developed PFC makes humans unique among mammals and is regarded as the brain’s executive which regulates the high level decision processes – both short and long-term [3, 4]. It is the PFC region that provides us the ability to build strategies to react to situations and to adapt these strategies to the environmental changes [5, 6, 7].

Functionally, the PFC is identified with several sub regions like ventromedial, dorsolateral and orbitofrontal areas, to mention a few. The orbitofrontal and ventromedial areas are involved in reward-based decision making processes by providing information regarding the decision’s attributes [8]. The role of dorsolateral PFC is to make decisions that call for the consideration of multiple sources of information [8]. One such region, which is of interest in this work, is the medial prefrontal cortex (mPFC).

While the executive role of PFC has been studied for a very long time, only recently has its contribution in memory tasks been studied. Imaging studies have shown activations of the PFC during memory-retrieval tasks [9, 10]. Behavioral studies have shown the role of mPFC in making responses that solely involves resolving conflicts/interference [11]. Behavioral studies in rats have shown the critical role of mPFC in resolving interference in memory-retrieval tasks [2]. This finding is also supported in studies that require rats to make choices from competing targets.

The lesioning of mPFC has been shown to cause impairments in the performance of rats in a Wisconsin card sorting task [12, 13]. In general, the rats in the card sorting tasks showed a
strong impairment in learning multiple contexts. When the rats had to learn the rules and respond to only one stimulus feature, they were unimpaired by the lesion. However, when the rats had to switch to a different stimulus response, they were impaired. The lesioning of mPFC impaired their performances in tasks that involved switching strategies [14, 15] and impaired their ability to learn items in a sequential order [16]. The mPFC has also been identified to be critical for various memory tasks that involve spatial memory [17, 18] and non-spatial memory [19].

Consistent with the above findings, Peters et al. (2013) demonstrated the significance of mPFC in a memory-retrieval task that involved selecting the correct response from other competing targets, i.e., resolving the conflict among multiple similar choices and making a final decision. They showed this role of mPFC by assessing the performance of the rats in a high-interference olfactory learning task entailing odor discrimination problems [20, 21]. In the odor discrimination problems, the rats were presented with two cups containing odorized digging medium of which only one cup had a reward (mostly food) buried inside. The rats were first trained to learn a list containing eight pairs of odor discrimination problems in a context. The rats were then trained to learn a second list (with eight pairs of odor discrimination problems) which had novel odors and previously used odors. This combination of novel and previously used odors induced interference.

Peters et al. (2013) performed three different experiments to conclusively show the role of mPFC in resolving interference. Their first experiment studied the effects of mPFC lesions in acquisition of only one item at a time [2]. They performed the first experiment in two arrangements. Both the arrangements were conducted with two groups of rats: Control rats with no mPFC lesions, muscimol rats with mPFC lesions.

- The first involved presenting one odor pair at a time until the rats reached a predefined criterion for that pair. The lesioned rats were able to learn all the pairs, each at a time, just as well as the control rats did.

- The second involved presenting all the odor pairs concurrently and training until the rats reached a criterion. The rats with mPFC lesions performed about 20 % worse than
control rats.

The results of the first experiment reflected that mPFC is involved in resolving interference caused by learning multiple items concurrently.

With this evidence, Peters et al. (2013) performed two more experiments to test the hypothesis that the mPFC contributes by providing contextual information to memory-retrieval tasks that involved interference. In this work, we present a computational model that embodied their (Peters et al. (2013)) behavioral results for the two experiments. The motivation of this work is to test their exact paradigm in a computational environment that involves only the hippocampus and the mPFC. Beyond that,

The goal of this thesis is to develop a model that can be used to predict results under different conditions without having to carry out the experimentation again.

To demonstrate the role of mPFC in a memory paradigm, we need a hippocampus model that performs the encoding and retrieval processes in addition to an mPFC model. We put forth a neural network based computational model of the hippocampus and the mPFC that represents this behavioral data. The biologically plausible hippocampus in the model developed performs an error-driven based learning during a given memory task [22] which is based on the Complimentary Learning Systems (CLS) neural network model of the hippocampus [23, 24]. For every single input presented to the hippocampus model during the encoding process, the hippocampus model produces an output pattern. The output pattern produced is based on the learning mechanisms built in the framework (detailed in chapter 2).

Unlike the biologically motivated hippocampus model, the mPFC in our model is need-based and non-biologically built. The goal of this research is to investigate if the mPFC plays a role in resolving conflicts by providing additional contextual information [2]. Peters et al. [2] performed context manipulations within the experimental conditions to show that the rats used this contextual information in a given task. The context manipulations involved using two different boxes, black and white, within which the rats were presented the odor discrimination problem. The rats encoded
this contextual information and associated it to the odor pairs. Similarly, during retrieval, the response is based on the context the stimulus is presented in. In our model, we frame it in a similar way by constructing a neural network layer with patterns of activations unique to every context. For instance, when the model is expected to learn in the context of a black box like the rats did, all the input items were combined with a specific pattern of activation that represented this context. This essentially gives us the same grounds as the behavioral experiment in the role of mPFC for the given task. Providing this contextual pattern of activation explicitly to the model creates the equivalence of the presence of mPFC in our model. The lesioning of the mPFC is done by not providing this contextual pattern of activation.

We simulate two out of the three experiments performed by Peters et al. (2013). The first simulation in our work involves demonstrating the effects of learning multiple conflicting items in absence vs. presence of mPFC. To investigate this, we train the model under two conditions.

- In the first condition, we present two lists (containing 8 pair of activation patterns) to the model. The first list and the second list are presented with mPFC. This serves as the control environment for our simulation.

- In the second condition, again, we present two lists to the model. The first list is presented without mPFC while the second list is presented in the absence of mPFC. For every pair presented in this list, half of the activation patterns are novel while the other half is a repetition from first list. This induces interferences.

This enables us to study if the absence of mPFC in the second condition created any impairment. Consistent with the experiment conducted by Peters et al. (2013), we find that the model without mPFC is impaired from learning the second list that consists interfering pairs. The previously learnt memories cause interference while acquiring new memories in the absence of mPFC. This interference leads to impairment in the performance.

In our second simulation, we investigate if the memories encoded without mPFC has any effects on the learning of new memories with mPFC. Like our first simulation, we train the model
under two conditions.

- The first condition involves the model in learning both the lists with mPFC. This is our control environment.

- The second condition involves the model learning the first list in absence of mPFC and learning the second list in presence of mPFC.

Again, consistent with the experiment conducted by Peters et al. (2013), we find that the model’s performance on learning the second list is better in the second condition than the first condition. After learning the first list in the absence of mPFC, the model is able to learn the second list (with conflicting items) fairly quickly. Unlike the previous simulation, the model in the second condition did not experience any interference.

In general, the results observed in these simulations are in agreement with the conception that PFC has a role in resolving interference [25, 26]. In our next chapter, we provide the details of the computational model and its learning framework. In chapter 3 and 4, we describe the methods and results of our first and second simulation respectively. Lastly, (in chapter 5) we discuss our results, key findings, limitations of the model and future work.
Chapter 2

Computational Model and its learning framework

Our research here involves studying about the role of mPFC in resolving interference during a memory retrieval task. We study this in a computational modeling environment. The computational set up is described below in three different sections of this chapter. First the learning framework that the computational model is built on and its underlying biologically plausible learning mechanisms are described. To demonstrate the memory retrieval task, we require a hippocampal model, which is described in the subsequent section. Lastly, we describe how we integrate the hippocampal model with mPFC to be able to test its role in resolving contextual interference in memory retrieval.

2.1 Leabra Learning Framework

The model is built on the Leabra (Local, Error-driven and Associative, Biologically Realistic Algorithm) framework with a balance between low-level, biologically detailed models, and more abstract computationally driven models [1]. As reflected by its name, Leabra is a combination of error-driven and associative learning. The learning dynamics of the framework are shown in Figure 2.1.

Each of the neurons is an Integrate and Fire neuron, a discrete spiking model, where each of them has a set threshold for firing. If the membrane potential of the neuron shoots over the set threshold, the neuron gets activated. The membrane potential of a neuron is calculated using a differential equation as shown in Equation 2.1 with 3 channels: e excitatory input; l leak current;
and inhibitory input \([1]\).

\[
\frac{dV_m(t)}{dt} = \tau \sum_c g_c(t) \bar{g}_c (E_c - V_m(t)) \tag{2.1}
\]

In being consistent with the electrophysiological method of describing, the overall conductance is decomposed into a time-varying component \(g_c(t)\). The conductance is then computed as a function of the dynamic state of the network while \(\bar{g}_c\) (constant value) controls the relative influence of the different conductance. All the layers in the models have bidirectional connectivity, which is vital for error-driven learning. The dynamics of this bidirectional connectivity is primarily managed by kWTA (k-winners-take-all) inhibitory function. They directly compute a level of inhibitory conductance \(g_i\) for all the neurons in a layer. The \(g_i\) value is placed so as to keep \(k\) units above their firing threshold (where \(k\) is a parameter set by the modeler, e.g. 15 %), with the remainder below threshold.

![Figure 2.1](image-url)

Figure 2.1: The figure shows the layers within the leabra framework with a focus on its primary component - integrate and fire point neurons and its learning mechanisms.

The learning rule used in competitive learning of mixture-of-Gaussians, a variant of Oja
normalization [27], is the hebbian learning mechanism in leabra [1]. The equation 2.2 shows the learning rule by which the weight changes of neurons occur in the hebbian environment:

$$\Delta_{\text{Hebbian}}_{ij} = y_i^+(x_i^+ - w_{ij}),$$

(2.2)

where the sending units of the neuron are represented as \((x)\) and receiving units of activation are represented as \((y)\).

The error-driven algorithm is the symmetric midpoint version of the GeneRec algorithm [28], which is functionally equivalent to the deterministic Boltzmann machine and Contrastive Hebbian Learning (CHL) [1]. The learning mechanism involves two key phases of activation called the minus phase and the plus phase. The minus phase is the stage where the network produces its output based on the learning rules. The plus phase is where the network is shown the target output for the given input. The learning rule using the contrastive Hebbian Learning is shown in the equation 2.3 with the plus phase and minus phase represented as the \(+\) and \(-\) respectively:

$$\Delta_{\text{ErrorDriven}}_{ij} = [x_i^+ y_j^+ - x_i^- y_j^-]$$

(2.3)

The ErrorDriven value is then subjected to a soft-weighted bounding to change the range between 0-1 through the equation shown below:

$$\Delta_{\text{ErrorDrivenSoftWeight}}_{ij} = [\Delta_{\text{ErrorDriven}} + (1 - ij) + \Delta_{\text{ErrorDriven}} - ij]$$

(2.4)

The error-driven and Hebbian learning components are combined additively at each connection to produce a net weight change using the equation:

$$\Delta W_{ij} = \epsilon[k_{\text{hebb}}(\Delta_{\text{Hebbian}} + (1 - k_{\text{hebb}})(\Delta_{\text{ErrorDrivenSoftWeight}})],$$

(2.5)

where \(k_{\text{hebb}}\) is a normalized mixing constant and \(\epsilon\) is a learning rate parameter.

The weight of all the neuron units and connections gets adjusted or changed based on the equation described above. All of these mechanisms are biologically driven and are considered to
mimic the biological neuron’s learning mechanisms.

2.2 Hippocampus Model

The hippocampal model used in the current work is built upon a series of structural and functional hypotheses based on anatomical and physiological data, which have been captured in the complementary learning systems (CLS) model of the hippocampus [23, 24, 22]. The hippocampal model is built in the Leabra learning framework as described in section 2.1. The complete computational model that was used for the simulations is shown in Figure 2.2 and a schematic model is shown in Figure 2.3.

The Entorhinal Cortex (EC) in the model is considered the cortical gateway to the hippocampus. This gateway feeds through the trisynaptic pathway (TSP) to the DG (Dentate Gyrus), CA3 (Cornu Ammonis), and then to CA1. Similarly, there is a parallel connection through the monosynaptic pathway (MSP) from the EC from and to the CA1.

The TSP connections via the perforant path from EC to DG and CA3 are broadly diffuse, and support the conjunctive binding of various distributed pieces of information into an overall episodic memory representation in the CA3. The CA3 has sparse and highly separable patterns of activity (which are further pattern-separated via the very sparse DG layer), resulting in substantially reduced interference from synaptic weight changes, thus enabling rapid learning of novel episodic or conjunctive information [23]. To recall when memories are cued; the recurrent connections in CA3 perform pattern completion to complete the cue.

The MSP between EC and CA1 is also essential for supporting memory retrieval. This pathway, unlike the TSP, is topologically organized and not diffuse. This organization is captured in the model by the simulated neurons in EC and CA1 into mutually interconnected slots. These are said to capture the different separable elements across all the cortical areas that converge on the EC [29]. Due to this topological arrangement, the MSP develops separable and invertible pathways where a given EC input pattern can be encoded over a sparser representation in the corresponding CA1 slot. This representation during retrieval gets transformed to retrieve the original input
activations.

Weighted connections within the MSP and between CA3 and CA1 utilize an error-driven learning signal that emerges from the firing dynamics of subfields within the theta oscillation [22]. These dynamics are modeled within the neural network as, for any given input pattern, three distinct time points of activation. The patterns of activation that arise from these three time points are contrasted to train the weighted connections within the MSP and the CA3 to CA1 connections citeKetzetal2013.

Building from the architecture described above, the subiculum is modeled as an intermediate processing layer between EC\textsubscript{in} and EC\textsubscript{out} shown with the red lines in figure 2.3. The role of the subiculum is to provide a novelty signal which is based on the mismatch of incoming stimuli, through the EC\textsubscript{in}, and the completed pattern of activation, from the EC\textsubscript{out}. The degree of mismatch determines how novel the presented item is. Intuitively, patterns that have not been well learned will produce essentially random patterns of output within the EC\textsubscript{out} layer, and will therefore produce a large mismatch with the input pattern. This mismatch will be reduced for well-learned patterns which, through training, have adjusted the weighted connections throughout the model to produce an EC\textsubscript{out} pattern similar to the input pattern in EC\textsubscript{in}.

The novelty measure, derived from the mismatch between EC\textsubscript{in} and EC\textsubscript{out}, in turn drives a dynamic learning rate in the connection from CA3 to CA1. The learning rate within this connection is proportionally decreased with the novelty signal, i.e., as novelty decreases so does the learning rate.

For every presentation of the input within the model, the error based on the difference in the input and output of the network is calculated using the Sum Squared Error. This error is normalized to a range between 0 and 1 and is referred to as Normalized Error in equation 2.6. If this error is higher than a threshold, referred to as Maximum Normalized Error, the novelty value is calculated on a linear scale between 0 and the Maximum Normalized Error value.
Figure 2.2: The figure shows the computational model that we used for all our simulations. The mPFC layer has connections (shown in red) to the Input layer where the activations of the mPFC are copied over to one of the columns of the Input layer. The input layer is the complete input to our model. This is fed into the EC_{in} (Entorhinal Cortex) as shown in black line. From EC_{in}, there are two major pathways: trisynaptic pathway (TSP, shown in green lines) and the monosynaptic pathway (MSP, shown in magenta). The TSP feeds to the DG (Dentate Gyrus), CA3 (Cornu Ammonis), and then to CA1. The MSP feeds to and from the CA1. The subiculum is modeled as an intermediate processing layer between EC_{in} and EC_{out} shown with the red lines.
Figure 2.3: The figure shows the Entorhinal Cortex (EC) in the model is assumed to be the cortical gateway to the hippocampus. This gateway feeds through the trisynaptic pathway (TSP) to the Dentate Gyrus (DG), CA3 (Cornu Ammonis), and then to CA1. Similarly, there is a parallel connection through the monosynaptic pathway (MSP) from the EC to the CA1 (and back). The subiculum is modeled as an intermediate processing layer between EC_{in} and EC_{out} shown with the red lines.
The novelty value is then used as a scaling parameter on a specific set of connection learning rates, and has a range from some set minimum value up to 1. This value is used as a multiplier on the default learning rate for connections from CA3 to CA1.

\[
Novelty = \begin{cases} 
\frac{NormalizedError}{MaximumNormalizedError} & \text{if Normalized Error < Maximum Normalized Error} \\
1 & \text{if Normalized Error > Maximum Normalized Error}
\end{cases}
\]

(2.6)

In summary, the hippocampus model will perform the encoding and retrieval processes in a memory task. The output of the mPFC, detailed in the section 2.3, is combined to the input of the hippocampus model.

**2.3 mPFC**

The model of mPFC in this work is not biologically motivated. The goal of the simulations is to demonstrate that the mPFC plays a role in resolving conflicts by providing additional contextual information [2]. Hence, in simulations when we require mPFC, we build our model to provide contextual information to the input. In the behavioral experiment by Peters et al. [2], they performed context manipulations within the experimental setup to show that the rats used this contextual information in a given task. The context manipulations involved using two different boxes, black and white, within which the rats were presented the odor discrimination problem. The rats encoded this information and associated this context to the odor pairs. Similarly, during retrieval, the response is based on the context the stimulus is presented in.

We built a layer in leabra framework, shown in 2.2, to replicate the role of the mPFC in providing contextual information to the hippocampus during encoding and retrieval. As it is used
in a lot of computational models, this information is only an abstract representation of contexts used in behavioral experiments like the color of the box. This contextual information binds the stimulus and the rewarded response for that stimulus. This layer is connected to the EC_{in} layer to provide the context whenever necessary. The activation pattern of this layer is copied over to the EC_{in} and combined with the original input that contains the two odor choices. The mPFC layer’s activations contribute to 25% of the total input activations. The details of the input are discussed in Chapter 3. On the whole, the input will be a combination of the stimulus input given to hippocampus and the contextual information to relate the stimulus with as received from the mPFC layer. This approach is not biological and computationally solely provides what the mPFC would have provided.
Chapter 3

The role of mPFC in acquisition of multiple conflicting items

Researchers have showed that the PFC region is active during memory retrieval tasks [25, 10]. Previous research has also proved that the mPFC, a sub region of PFC, plays an active role in resolving conflicts while responding to stimuli [11]. A recent behavioral study by Peters et al. (2013) showed that the absence of mPFC does not impair the ability of rats to learn one item at a time even if those items have conflicting choices [2]. However, the study also showed that the absence of mPFC impaired the ability to learn multiple conflicting items. The behavioral experiment was carried out with rats. It involved the rats in learning two lists (List 1 and List 2 as referred from here on) of odor discrimination problems [2]. At each trial, the rats were presented with a pair of cups containing the odorized bedding, of which one contained a rewarding food item and the other contained a non-rewarding item. The rats had previously been trained to dig inside the cups and retrieve the rewards. For every pair of odor cups presented, the rats had to make a choice (amongst the two cups) that contained reward. The experiment was performed with two groups of rats: Control and Muscimol. Below shows the details of the two lists and the environment the two groups of rats were trained on.

- Control Rats
  
  * List 1 (contained 8 pairs of odors) was presented in the presence of mPFC, until they reached a predefined criterion (100% accuracy).
  
  * List 2 (contained 8 pairs of odors) was presented in the presence of mPFC for 5 days.
• Muscimol Rats

  * List 1 (contained 8 pairs of odors) was presented in the presence of mPFC, until they reached a predefined criterion (100 % accuracy).

  * List 2 (contained 8 pairs of odors) was presented in the absence of mPFC for three days. No infusion (lesions) were given during the last two days. The last two days when there were no infusions given. This was done to observe if the recovering of mPFC had any effects.

For List 2, only half of the odors were novel odors while the other half were repeated odors from List 1. This was done purposely to induce interference [30, 2]. Half of the repeated odors were the ones that were rewarded previously and half of them were not rewarded previously. This was done to ensure the rats did not adopt any strategy in making a choice of reward [30, 2]. The results, in general, showed that the absence of mPFC impaired the rats from learning multiple conflicting memories.

With these findings from Peters et al. (2013), we simulate their experiment, in a computational framework, to see if our results agree with theirs.

### 3.1 Simulation Method

While preserving the concepts from the behavioral experiment, we set up our simulation methods in the computational model as described in this section. All the layers of the model are made of units of neurons. A set of 7 by 7-square units of neurons is referred to as a slot. The network used consists of 500 units of DG and 100 units of CA3. The CA1 layer consists of 8 slots, where each slot is a set of 10 by 10 square units of neurons. The model is initialized with a random set of weights between all connections before the start of the simulation. The details of the framework in which the computational model is built are described in Section 2.1, 2.3 and 2.4 in Chapter 2. Figure 3.1 shows the input format in which the input is fed into the model. The input to the model (specifically, the EC_in layer) as shown in Figure 3.1 consists of four sub items:
• two choice items (red box), pair of odors in the behavioral experiment,

• right rewarded choice of the two items (black box) and

• the contextual information that the mPFC layer produces (blue box).

The activation patterns of the slots were generated randomly. Each of the four items is represented in the input layer as a set of two slots where only one slot is a unique slot and the other is a copy. The reason for building the sub items in this fashion is the biological construction of the hippocampus, specifically the trisynaptic pathway within the hippocampus. This duplication serves as an excitatory input to CA1 and the Schaffer collaterals to detect and react to novelty in the environment [31, 32]. This structure of input pattern is fed into both the trisynaptic and monosynaptic pathway of the hippocampus which produces the output to the $EC_{out}$ layer as described in Chapter 2.

For this simulation, the model is given two lists (List 1 and List 2) with each item in the lists in the input format as described earlier in this section. Each list is trained for a total of 10 epochs. A single epoch implies a single presentation of each of the items in a given list. Similar to the behavioral experiment, to induce interference or conflicts in the choices, half of the choices in the List 2 were novel choices while the other half were previously presented choices in List 1. Of those repeated choices, half of the previously rewarded choices from List 1 were presented as non-rewarded choices in list 2. Likewise, half of the previously non-rewarded choices from List 1 were presented as rewarded choices in List 2. We perform our simulation in two cases. The difference in the two cases is training the model in either the presence or absence of mPFC. Table 3.1 provides the information on the differences. Analogous to the behavioral conditions, we choose to train our model in the same ratio of 3:2 for lesioned and non-lesioned cases. These correspond to 6 epochs and 4 epochs in our simulations. It is to be noted that although an epoch means a single presentation of the item to the model, it does not translate directly to presenting each item once to the rats. The reason being, the model’s framework has built in learning rates for the connections to learn faster in time (weight changes are faster). The process by which the learning rate modulates the weight changes
Figure 3.1: The figure shows the input format in which input was fed to the computational model. The layers are built of units of neuron (shown in black small box). A set of 7 by 7 neuron units is called a slot. Each column (two slots together) represents one sub item of our input. The pair of choices is the first two columns (shown in red). The right-rewarding choice for that pair is the column shown in block. The activations of mPFC are carried over from the mPFC layer to the blue column in the input.

is described earlier in Equation 2.5 in Chapter 2. This provides increased efficiency in simulating real-time experiments. Based on previous analysis of the model’s efficiency, this number of epochs were chosen [22].

<table>
<thead>
<tr>
<th>List 1</th>
<th>List 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 Epochs</td>
<td>4 Epochs</td>
</tr>
<tr>
<td>6 Epochs</td>
<td>4 Epochs</td>
</tr>
<tr>
<td>Case A</td>
<td>mPFC</td>
</tr>
<tr>
<td>Case B</td>
<td>mPFC</td>
</tr>
</tbody>
</table>

Table 3.1: The details of the two cases and their training environments under which the simulation 1 were performed.
We then plot our results in the same format as in the behavioral experiment, showing a learning curve of the acquisition of List 2 items in both conditions: with and without mPFC. The simulation is repeated 6 times as the behavioral experiment performed the experiment with 6 rats. The behavioral experiment performed multiple trainings of the same list in one day and represented the final training accuracy at the end of the day. Similarly, in our results, each data point is the accuracy after training for 2 epochs.

3.2 Results and Discussion

During the testing phase, the model is cued with both the choice and the contextual information (if there was no mPFC lesion). The correctness of the response is measured by whether the model made the right choice for that pair, i.e., it reproduces the patterns of activations that represent the right choice for that given pair and context. The results of acquisition of the List 2, under Case A and Case B, for our simulation is shown in the Figure 3.2 and the results of the experiment performed by Peters et al. (2013) are shown in Figure 3.3. The method used to measure the correctness is called the name error method, which can directly be related to match our requirement. The technique needs to measure whether the model made a right choice by producing a pattern of activation that is either exactly the same as the original right choice or a closer match to the original choice. A closer match will imply the bias of the model to choose the option in the event of not making a discrete final decision and hence is regarded as a right choice. The name error procedure precisely makes such measurements with respect to the closeness of the choice, i.e., if the choice made is closer to the right choice, and then the name error is zero. The error bars represented in the Figure 3.2 are the regular SEM (standard error of mean).

Since the behavioral experiment showed the role of mPFC in resolving the interference was by providing contextual information, in our simulations, we perform with two different conditions. One with mPFC, i.e., providing contextual information in the input and the other without mPFC, i.e., not providing contextual information in the input. The contextual information provided to the second list is different from the contextual patterns of activation that was provided to the List
1. The results of the simulation and the behavioral experiments as seen from the Figure 3.2 and Figure 3.3 respectively are summarized in Table 3.2. Table 3.2 shows the % correct of List 2 at the end of training of 6 epochs (in our simulation) and 3 days (in behavioral experiment).

<table>
<thead>
<tr>
<th></th>
<th>Simulation Results</th>
<th>Behavioral Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case A</td>
<td>81.25 %</td>
<td>85 %</td>
</tr>
<tr>
<td>Case B</td>
<td>43.75 %</td>
<td>55 %</td>
</tr>
</tbody>
</table>

Table 3.2: Table shows the % correct of List 2 at the end of training of 6 epochs (in our simulation) and 3 days (in behavioral experiment).

Clearly, the performance of the model in our simulations under Case B performs lesser than the model under Case A. Note that, in both cases, before training on List 2, the accuracy of List 1 reaches 100 %. Let us consider the Case B in little more detail. When the model is trained on List 1 (in the presence of mPFC), the ability to acquire multiple conflicting items is not affected. It reaches
Figure 3.3: Percent correct data are shown for the final day of training on list 1 (Last) and during the five training sessions (first three days with mPFC lesions, muscimol rats and last two day with no lesions, control rats) of list 2, for control (open symbols) and muscimol (filled symbols) rats, and for the different context (squares) and same context (circles) rats. Figure adopted from [2]

<table>
<thead>
<tr>
<th></th>
<th>Simulation Results</th>
<th>Behavioral Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case A</td>
<td>97 %</td>
<td>98 %</td>
</tr>
<tr>
<td>Case B</td>
<td>93 %</td>
<td>90 %</td>
</tr>
</tbody>
</table>

Table 3.3: Table shows the % correct of List 2 at the end of training of 10 epochs (in our simulation) and 5 days (in behavioral experiment).

to full accuracy by 10 epochs of training. Note that, the items here are encoded with the contextual information and each item is internally associated with this context. When the model is presented with List 2, in the absence of mPFC, the contextual information is not provided. Since, we have items in List 2 that are repeated from List 1, the model needs a method of differentiating these
new items. A key information that can be used to differentiate is missing, i.e., the context/mPFC. This results in impairment in the ability to acquire new memories from List 2. Based on these results, we suggest that mPFC is playing a significant role in conflict resolutions. The hypothesis now is that the role of mPFC is to provide contextual information that the items can be associated with. If this is possible, then providing this contextual benefit to Case B, after 6 epochs of deprival, should increase the performance. Since the mPFC is not lesioned at all in the Case A, it shows a consistent improvement in its performance even for the last 4 epochs too. However, an interesting observation is in Case B, when mPFC was lesioned. We observed that the model’s performance improves considerably to 93.0 % from 43.75 %. There is a substantial improvement in learning when the contextual benefit is provided. This suggests that the interferences experienced by the model are resolved by the context that mPFC provided. Table 3.3 shows a comparison of the % correct of list 2 at the end of training of 10 epochs (in our simulation) and 5 days (in behavioral experiment).

To measure the interferences posed, we calculate an Interference Index that indicates the interference experienced by our model. This is calculated as the % of pairs correct on list 2 (after 6 epochs) minus the % of pairs correct on list 1 (after 6 epochs).

Figure 3.4: The interference index shows the decline of performance in learning list 2 after learning list 1. The Interference index is calculated as the % correct on list 2 minus the % correct on list 1 for the two conditions, with and without mPFC.
Figure 3.4 shows the interference index for both the control and lesion cases. The case without mPFC experiences more interference than the condition with mPFC, again confirming the hypothesis that mPFC plays a significant role in reducing interference by providing additional contextual information.

On the whole, after the complete training in the two different cases, our simulation results show that the performance of the model with the contextual benefit is 6% higher than the performance of the model without contextual benefit. However, the original experiment show a difference of 10% across both conditions. These differences could be caused by several reasons that we address in Chapter 5.

In summary, the results from our simulation suggest the following conclusions.

- The memories learnt with mPFC pose proactive interference on any new conflicting memories that needs to be learnt.

- To resolve this proactive interference, mPFC is required.
Chapter 4

The contributions of mPFC during encoding

Our first simulation results showed that the absence of mPFC impaired the learning of multiple conflicting items. We saw the model’s performance was better when mPFC was active in the form of contextual information. We observed that the memories which are previously learnt in the presence of mPFC interfered with acquisition of new memories in the absence of mPFC.

The goal of this simulation is to see if the previously learnt memories in the absence of mPFC interfered with the acquisition of new memories in the presence of mPFC. Previously, we saw the immediate effects of inactivating mPFC while learning the List 2. Here, we inactivate mPFC before learning the List 1 to see if it had any long-term effects.

4.1 Simulation Method

We set up our simulation to match the behavioral experiment. The input to the model is in the same format as we had in our previous simulation (shown in Figure 3.1). The input consists of four sub items:

- two choices,
- the choice that is rewarding (one of the two choices picked randomly) and
- the contextual information.

For this simulation, the model is given two lists (List 1 and List 2) with each item in the list in the format as described earlier in this section. To induce interference or conflicts in the choices,
half of the choices in the List 2 are novel items while the other half are previously presented choices in List 1. Out of the repeated choices, half of the previously rewarded choices from list 1 are presented as non-rewarded choices in list 2. Similarly, half of the previously non-rewarded choices from list 1 were presented as rewarded choices in list 1. Similar to our first simulation, the first list here is trained until the items reach a 100% accuracy. It took 20 epochs for the first list to reach 100% accuracy. The second list, however, is trained for 10 epochs. Each of the list’s training set is split into two groups of epochs (like our first simulation) in the ratio of 3:2. This simulation is also executed in two cases. The two cases under which we ran our simulations are:

<table>
<thead>
<tr>
<th></th>
<th>List 1 - 20 Epochs</th>
<th>List 2 - 10 Epochs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case A</td>
<td>mPFC</td>
<td>mPFC</td>
</tr>
<tr>
<td>Case B</td>
<td>No mPFC</td>
<td>mPFC</td>
</tr>
</tbody>
</table>

Table 4.1: The details of the two cases and their training environments under which the simulation 2 were performed.

We observe the acquisition pattern of List 2 under both Case A and B. Case A serves as our control environment. The behavioral experiment, by Peters et al. (2013), performed multiple trainings of the same list in one day and represented the final training accuracy at the end of the day. Similarly, in our results, each data point is the accuracy after training for 2 epochs.

4.2 Results and Discussion

Our results showing the acquisition of List 2 in the two different cases are depicted in Figure 4.1. The line with square markers represents Case A, the control case. It shows the acquisition of the List 2 after learning the List 1 with mPFC. The line with round markers represents Case B and shows the acquisition of the List 2 after learning the List 1 without mPFC. The error bars represented in Figure 4.1 are the regular SEM (standard error of mean). Each point is a representation of the accuracy after training for 2 epochs. Each data point is an average of six simulation runs (as the behavioral experiment performed the experiment with 6 rats).

Figure 4.2 shows the acquisition of the List 2 in rats in both Case A and B from the behavioral
Figure 4.1: The acquisition of list 2 after learning the first lists under presence and absence of mPFC. The error bars represented in the figures are the regular SEM (standard error of mean). The line with square markers represent the learning of list 1 in control environment (with mPFC) and the lines with round markers represent the learning of the list 1 in lesioned environment (without mPFC). Each point is the accuracy after training for 2 epochs. Also, each data point is an average of six simulation runs.
Figure 4.2: Percent correct for control and muscimol rats during list 2 acquisition. The rats that had inactivation of mPFC (when learning list 1) perform better on learning list 2 than the rats that had no inactivation. Figure adopted from [2]
Table 4.2 summarizes the results of the simulation after 10 epochs. Since in both cases, List 2 is fully trained in presence of mPFC, the table reports the accuracy at the end of 10 epochs.

<table>
<thead>
<tr>
<th></th>
<th>Simulation Results</th>
<th>Behavioral Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case A</td>
<td>91.6 %</td>
<td>93 %</td>
</tr>
<tr>
<td>Case B</td>
<td>95.8 %</td>
<td>96 %</td>
</tr>
</tbody>
</table>

Table 4.2: Table summarizing the results from our simulation and behavioral experiment. It shows the % correct of List 2 at the end of training of 10 epochs (in our simulation) and 5 days (in behavioral experiment).

In general, the acquisition percentages of List 2 in Case B is higher than the acquisition percentage of List 2 in Case A. In Case B, (after learning the first list in absence of mPFC), after 10 epochs, the accuracy reached 95.8 %. However, in case A (after learning the first list in presence of mPFC), the second list reached 91.6 % accuracy. It is to be noted that the difference at the end of 10 epochs in both cases is not large. However, the key observation is that the learning curve of Case A is always lower than Case B.

Consistent with our results, On the acquisition of List 2, the behavioral experiment showed that the rats in Case A were approximately at 93 % accuracy, while the rats in Case B were at 96 % correct.

To better understand the results, Let us consider Case B in more depth. The items in List 1 are learnt in absence of mPFC. The memories of List 1 are encoded without any contextual information. When the List 2 is presented, contextual information is provided. The model clearly has additional information to associate the choices presented to it. This additional information helps the model to learn the items of List 2 accurately.

Now, one might ask, if providing additional information helps the model in learning a list quickly, why the model doesn’t learn List 2 in Case A with as much accuracy? In Case A, When List 1 is learnt with context, the representations of the items are bounded internally with the contextual information. However, when List 2 is presented with a different context, the model
actually experiences an increased interference. mPFC is critical in encoding memories, but at the same time, persistent intrusions of these memories pose interference while learning new memories. The behavioral experiment also observed the same results and in supportive of this result, studies have shown that the mPFC is active even during memory encoding [35, 36].

4.2.1 Observations under a modified environment

Since, we observe that the results from both are simulation are consistent with the results from Peters et al. (2013), we significantly believe our model to be a good representation of their results. With that, we make a modification to the simulation set up. This modification is to test the effects of increasing the interference by not introducing any new odors in the List 2. In our earlier simulation, only half of the odor items in the List 2 are repeated from the List 1. Here, we repeat all the odors and all the previously non-rewarded choices now become rewarded choices. This gave rise to a complete reversal in the right choices from List 1 to List 2. By not introducing any novel items, there will be an increased interference. Hence, we expect that the acquisition of list 2 to be difficult than our previous case.

The model is trained under the same cases (Case A and Case B) as mentioned earlier. The learning of the List 1 is performed either in presence (Case A) or absence (Case B) of mPFC while the learning of the List 2 is always in presence of mPFC.

The results of the simulation are shown in Figure 4.2.1. The results are also summarized in Table 4.3.

<table>
<thead>
<tr>
<th></th>
<th>Simulation Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case A</td>
<td>85 %</td>
</tr>
<tr>
<td>Case B</td>
<td>95 %</td>
</tr>
</tbody>
</table>

Table 4.3: Table summarizing the results from our modified simulation environment. It shows the % correct of List 2 at the end of training of 10 epochs.

Under these modifications, we observe that the performance of the model on List 2 in Case B (when the items of List 1 are learnt without mPFC) at the start is lower than the performance
Figure 4.3: The percentage correct of List 2 after learning the first list under presence and absence of mPFC. The items in the second list in this case were repeated full from first list and no novel items introduced. The error bars represented in the figures are the regular SEM (standard error of mean). The line with square markers represent the learning of List 1 in control environment (with mPFC) and the lines with round markers represent the learning of the List 1 in lesioned environment (without mPFC). Each point, here again, is an accuracy after 2 epochs.
of the model in Case A. After 2 epochs, the performance of Case B increases drastically and stays consistently higher than Case A. In this situation, the items learnt in List 1 under Case A provide proactive interference when new items are acquired. When all the items in List 2 are not new, the model in the Case A experienced an initial increase in interference since all the items are previously learnt.

However, we see that at the beginning, Case A is better than Case B. Based on our previous conclusions, we propose a possible reasoning for the behavior of Case A at the start of learning List 2. In Case B, at the beginning, the model is presented with old memories with a context that was not presented earlier. The model possibly starts to associate its old memories to the context, while later the model, based on the feedback provided, the model starts to encode those items as new memories. Hence the performance shows an improvement from there on.

To summarize, the inferences that we can derive from the results of the simulations presented in this chapter are:

- **The memories encoded without mPFC does not pose any interference in acquisition of new conflicting memories.**

- **The mPFC is not only critical in retrieval of conflicting memories, but also in encoding of conflicting memories.**
Chapter 5

Discussion

The PFC region in the brain has been identified with multiple functions such as regulating the high-level decision processes, building strategic plans and adapting those strategies to changes in the environment [3, 4, 5, 6, 7]. These functions of the PFC region have been long studied. The PFC, more specifically the mPFC, is being extensively studied in the recent years for its contributions during a memory-retrieval task. The mPFC is thought to play a critical role in memory-retrieval tasks that involve conflict resolutions. In support of this hypothesis, studies have also shown the interaction of mPFC and the hippocampus in memory processes [33, 34]. Peters et al., 2013 showed that the mPFC resolves conflicts that arise out of interferences during encoding and retrieval of memories [2].

The motivation behind this thesis was to build a neural-network based computational model that represented the results from their behavioral experiments' results. In our first simulation results, we observed the interferences that the learning of memories with mPFC poses on new memories that were learnt later with mPFC. We also observed how this interference affected the performance of the model in learning new memories.

Having seen the interferences the memories encoded with mPFC poses, in our second simulation, we investigated the interferences caused by the memories that were encoded without mPFC. We found that the memories that were encoded in the absence of mPFC did not pose any interference on new conflicting memories. As a consequence of this absence of interference, the model was able to acquire the new conflicting memories faster. Both our simulation results showed impairment
in performance while conflicts have to be resolved without mPFC. In both encoding and retrieval, the performance was impaired.

Previous modeling results also support our observations and provide a mechanism that could be involved in resolving such competitive targets [37]. Similar and supportive to our modeling technique, experiments have shown that the mPFC is possibly connected with these memory representations through its interaction with hippocampus. During retrieval of the memories that triggers several possible targets, the connections with mPFC provides a solution by resolving these possibilities with contextual information previously encoded [38, 39]. Studies have also shown biological connections between mPFC and hippocampus in humans and animals [40].

The percentages of the results we observe in our simulations do not match the exact percentages of biological experiment performed by Peters et al. (2013). These differences can be caused by several reasons such as,

- the non-biological mPFC layer that only imitates the role of mPFC.
- learning rates of the connections in the model and
- the number of epochs chosen.

One key limitation to our model is the mPFC layer. We do not have a biologically motivated layer for this simulation. As mentioned previously, our layer is only an imitation of mPFC’s role and not the mPFC itself. mPFC encodes and provides additional contextual information to the model to bind the stimulus and the response. Similar to previous computational researches, the contextual information is just an abstraction of the contextual manipulations like the color of box used in the behavioral experiments [50]. We only have modeled this aspect computationally, i.e., in cases when mPFC is present; we provide contextual information to the model along with the input. This approach however, cannot provide answers to biological questions related to mPFC such as, how this contextual information is generated, what triggers this contribution.

A customary criticism as in all computational models is the real need for a model that represents the experimental data. The model here represents the contribution of mPFC in conflict
resolution based on previous behavioral data. It often appears that the computational model is built in such a fashion that it can explain the data that we base the hypothesis on. Although to an extent, we fit the model by adjusting our free parameters, which represents the data the best. Whenever there are free parameters, fitting is always involved. We need to be careful is that the model is not over fitting the data. Over fitting the data can also lead to poor predictions. Avoiding over fitting is achieved by limiting the capacity of the model. For instance, in our model, we limited the size of the layers we used and the number of epochs. The size of the layers is set based on our previous modeling simulations [22]. We also set the number of epochs based on the capacity of the hippocampus that we explored earlier in [22].

5.1 Context representation in mPFC vs. Hippocampus

The hippocampus is also renowned for generating contextual memories, especially, in the mechanism of associating the learnt items with a context that provides an automatic process for a conflict-free retrieval [41, 20]. However, there is a difference in the contextual representations that the mPFC and the hippocampus form. The hippocampus is well-known for spatial and contextual processing [42, 43, 44]. The mPFC contributes to various cognitive processes that are context-based [17] and contextual fear conditioning [45]. The functions of the mpFC however, are said to be dependent on reciprocal interactions with the hippocampus [46, 47]. To explain the difference in both their representations, let us consider a space navigation task in a maze performed by rats. The representations that are formed by the hippocampus capture the exact location of the rats within the environment [2]. However, the mPFC only captures a more global representation of this environment and its contexts and does not capture a complete spatial map [48, 49]. In other words, the mechanism in hippocampus is only key when the item in memory is linked to a specific context. The mPFC, as seen in our simulations, exerts a top-down influence on retrieval process when there is a conflict during retrieval by using a more general interference resolution mechanism [2].
5.2 Future Work

Beyond just representing the behavioral data, our goal for building the computational model is to use this model in the future to predict results under a different condition. The general motivation to build such models is that they can provide an explanation to very detailed cognitive processes in the brain. The current simulations results suggest that the mPFC interacts with the hippocampus during a memory process involving interference. We also find that the interaction is primarily by providing contextual input that represents the context in which the memories are shown. However, there are various other ideas that can be explored with this model.

A potential field to explore is to see what other cognitive processes involve or require this contextual input that comes from mPFC. A computational model that studied about context-dependent encoding of fear and extinction memories in basal amygdala suggested that it could be the mPFC that provides the contextual input during the process [51]. In supportive of this claim, anatomical explorations have shown projections from mPFC on to the basal amygdala [52]. Similar to our approach of providing context in the form of an additional input, in [51], they also provided an additional information along with the input that represented the contexts.

An assumption of our model is that the role of mPFC is only to provide contexts in resolving conflicts. This also leads another proposition that we could explore using our model. Numerous computational models have shown the role of mPFC as a performance monitor [53, 54]. Few computational models explored this in detail and showed that the mPFC gets recruited and trained by error signals to respond to stimuli that preceded an error [55]. When the stimuli is seen again, the response cues that are associated with higher probability of error drives for more activations of the mPFC cells [56].

5.3 Conclusions

In this research, we present a computational model that is representative of a recent behavioral study by Peters et al., (2013). The computational model demonstrates the critical role
and need for mPFC during a memory process that involves conflicts resolution. We show that the mPFC resolves the conflicts by providing a contextual representation along with the input to the hippocampus. Our results also illustrate that the mPFC was critical in learning multiple conflicting items. The results confirmed that the memories that are encoded in the presence of mPFC exerted interference to any new conflicting memories that are presented. In general, the results observed in our research were in agreement with several other research findings such as, the conception that PFC has a role in resolving interference in general [25, 26].

We also presented the limitations of our model such as the simplified imitation of the presence of mPFC. Though it has limitations, the model was a good representation of the behavioral experiment we aimed to replicate. Lastly, We also proposed ideas that can be explored further using this model in various dimensions such as, the contributions of mPFC in contextual fear conditioning.
Bibliography


