Self Organising Associative Representation Learning Model

Esther Mutanu Mulwa

supervised by:

Dr. Esther Mondragón

Prof. Eduardo Alonso

A thesis presented for the degree of

Doctor of Philosophy

City St George’s, University of London

Department of Computer Science

School of Science & Technology

July 2025

Declaration

I, Esther Mutanu Mulwa, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Table of Contents

List of Figures

Abstract

We present a self-organizing, real-time associative learning model that incorporates a biologically plausible image processing mechanism. This model offers a naturalistic approach for generating hierarchical perceptual representations of visual stimuli built from the convolution and pooling layers of a Convolution Neural Network (CNN) model. We showcase the implemented technical aspects used in building this model, as well as the algorithms developed for the operations driving different computations. We demonstrate that this model can account for most characteristic associative learning phenomena, fully conceptualizing cue competition as encapsulated by a global error correction mechanism. The integration with CNN endows the model with an elemental framework from which we present strong evidence of the formation of complex representations during learning and provide the necessary computational and algorithmic approach for how compound stimuli are formed, offering a natural mechanism to represent compound stimuli and discriminate them from their constituent elements. Critically, the model, elemental in nature, is capable of representing non-linearity, accounting for non-linear discrimination without the need to postulate *ex nihilo* stimulus representations.

List of Abbreviations

AA	Associative Activation
CNN	Convolutional Neural Network
CR	Conditioned Response
CS	Conditioned Stimulus
DDA	Distributed Associative Architecture
ISI	Inter-Stimulus Interval
ITI	Inter-Trial Interval
OA	Overall Activation
RF	Receptive Field
SOARN	Self-Organizing Associative Recurrent Network
UR	Unconditioned Response
US	Unconditioned Stimulus
V	Associative strength value
λ	Dynamic asymptote of learning
δ	Temporal spread parameter/standard deviation
ν	Asymptote weighting parameter

Glossary of Terms

Acquisition The process by which a neutral stimulus becomes associated with an outcome through repeated pairings.

Blocking A phenomenon where prior learning about one cue prevents or "blocks" learning about a second cue when both are presented together.

Conditioned Inhibition A form of learning where a stimulus signals the absence of an expected outcome, acquiring negative associative strength.

Extinction The gradual weakening of a conditioned response when the CS is repeatedly presented without the US.

Negative Patterning A non-linear discrimination task where subjects learn to respond to individual stimuli when presented alone (A+, B+) but must inhibit responding when those same stimuli are presented together as a compound AB-). This task is a benchmark test for configural learning because it cannot be solved through simple elemental summation. The compound must be processed as a unique configuration distinct from its individual components.

Element A computational unit representing one spatial location in a receptive field that can serve as both predictor and outcome in the associative network.

Predicted Receptive Field The learned internal representation that emerges when a stimulus activates its associated outcomes through the network.

Receptive Field A 56×56×4 feature map produced by the CNN, capturing the visual features of a stimulus at different spatial scales.

Chapter 1

Introduction

1.1 The Phenomenon of Visual Stimuli Representation

Many mechanisms underlie the cognitive function of performing visual tasks. For example, when we see images, a subset of neurons in the visual cortex are co-activated. Each active neuron encodes abstract stimulus features such as shape, colour, corners, textures, or edges. As a result, neural activities are distributed around various regions of the brain, collectively forming abstract representations of the images we see. However, a full representation of a stimulus does not rely solely on the current input. Complex stimulus representations often include elements of different and past sources correlated with the current stimulation [1]. One mechanism that has long been postulated to contribute to the formation of complex event representations is associative learning [2, 3, 4]. Thus, in addition to specific perceptual mechanisms, this thesis focuses on associative mechanisms that underlie the formation of stimulus representation. Concurrent neural activities form links between neurons, mainly supported by the relation of events containing stimuli with information about other cues. This framework conceptualizes how the perceptual system may work under associative mechanisms and forms a basis for the motivation of the work presented in this thesis. Our research work focuses on building a real-time associative learning model that simulates the involvement of associative learning mechanisms within the formation of complex visual stimuli representation as extracted from a perceptual system.

1.2 Visual Processing and Hierarchical Organization

The visual cortex is generally organized into an interactive hierarchy of interconnected areas that pass information to its immediate neighbours. The flow of information is bottom-up, where sequential information moves exclusively from lower regions to higher ones, and top-down, with feedback connections loops that enable the circulation of information among the visual areas [5]. For the Images we see, processing starts with a primal sketch and concludes with a high-level representation such that we present partial cues of the image, and the brain can retrieve the learned representation of the object.

Convolution neural networks (CNN) exhibit functionalities similar to those of the visual cortex. Deep CNN layers were used to implement the perceptual system, performing visual processing tasks such as detection and exploring regularities in image patterns. In our model, CNNs play a fundamental role in extracting the perceptual representation of the visual stimulus as filter maps, which are used as input stimuli for events designed to simulate associative learning mechanisms. This model has been implemented and built by integrating deep Convolution Neural Network (CNN) layers within the associative learning model.

1.3 Review of Formal Models of Associative Learning

1.3.1 Foundational Models and Their Contributions

The field of associative learning has developed through successive theoretical frameworks, each attempting to capture different aspects of how animals learn relationships between events [6, 7, 8]. We review these models to establish the theoretical context for our work and identify the limitations our model addresses [9, 10].

1.3.1.1 The Rescorla-Wagner Model (1972)

This foundational model [11, 12] formalized learning as a function of prediction error, proposing that associative strength changes proportionally to the discrepancy between expected and actual outcomes

$$\Delta V_i = \alpha_i \beta_j (\lambda_j - \sum V_{\text{present}})$$

The model successfully accounts for blocking and overshadowing phenomena through its global error term. However, it cannot adequately represent configural stimuli without additional assumptions, fails to account for within-compound associations, and does not explain recovery phenomena such as spontaneous recovery and renewal.

1.3.1.2 Pearce's Configural Theory (1987, 1994)

In contrast to elemental approaches, Pearce [13, 14] proposed that animals learn about entire stimulus configurations rather than individual elements. The associative strength of a configural element C is given by:

$$V_C = E_C + \sum S_{Ci} E_i \tag{1.1}$$

where E_C is the direct associative strength and S_{Ci} represents a generalization based on similarity. Although successful in accounting for negative patterning and configural discrimination learning, the model inadequately addresses generalization between similar stimuli and scales poorly to realistic stimulus environments.

1.3.1.3 McLaren and Mackintosh's Elemental Model (2000, 2002)

This model attempts to bridge elemental and configural approaches by proposing that stimuli are represented by large sets of micro-elements. Through coactivation, configural units form automatically, and salience is modulated through associative processes. However, the model struggles with computational complexity and lacks a clear mapping to neural mechanisms.

1.3.1.4 Temporal Difference Learning

Developed from reinforcement learning techniques, these models extend classical conditioning phenomena to incorporate temporal dynamics. While powerful for modeling timing effects, they typically require discretized-time representations and struggle with simultaneous compound discriminations.

1.3.1.5 Double Error Dynamic Asymptote Model (DDA)

The DDA model [10] represents a recent advanced model of associative learning, introducing a dynamic asymptote based on the distance between predictor and outcome activities. Unlike models with fixed asymptotes, the DDA implements a variable learning parameter:

$$\lambda^t_{i,p\to j,o} = \frac{\hat{A}^t_{j,o} - |\hat{A}^t_{j,o} - \hat{A}^t_{i,p}|}{\max(\hat{A}^t_{j,o}, \hat{A}^t_{i,p})} \tag{1.2}$$

The model incorporates double error terms, one for outcomes and one for predictors, enabling within-stimulus learning alongside traditional stimulus-outcome associations. This framework successfully accounts for a wide range of phenomena while maintaining computational efficiency. Critically for our work, the DDA's elemental architecture and real-time processing capabilities make it particularly suitable for integration with perceptual systems.

1.3.2 Key Phenomena Tested

Several fundamental associative learning phenomena were selected to validate our model.

Acquisition and Extinction: The formation of associations between stimuli (A+) and their subsequent weakening when the outcome is removed (A-).

Blocking: Prior learning about one cue (A+) prevents or reduces learning about an added cue (AB+), demonstrating cue competition.

Conditioned Inhibition: A stimulus (X) learns to suppress responding when paired with an excitor in the absence of the outcome (A+, AX-).

Negative Patterning: The ability to discriminate between individual stimuli that predict an outcome (A+, B+) and their compound that does not (AB-).

These phenomena test whether our integrated model can account for basic associative processes, cue competition, inhibitory learning, and non-linear discrimination—core challenges for any comprehensive learning theory.

1.4 Limitations of Existing Approaches

Current associative learning models face several key limitations. Existing models typically use abstract stimulus representations rather than processing actual visual inputs. This abstraction creates a gap between how perceptual systems work and how learning occurs. Additionally, elemental models like Rescorla-Wagner cannot account for non-linear discriminations, as they assume simple summation of associative strengths.

Most models also lack clear connections to neural mechanisms. Although they successfully predict behavioral phenomena, they do not specify how their proposed computations might be implemented in the brain. This limits their explanatory power and biological relevance.

The requirement for prespecified stimulus representations is particularly problematic. Models assume that stimuli have already been parsed into an appropriate representation or configuration, without explaining how these representations emerge from stimuli input. This is especially limiting when considering complex visual stimuli.

Finally, many models become computationally intractable as the number of stimuli increases, limiting their applicability to realistic learning scenarios. These limitations motivated our approach of integrating visual processing mechanisms with associative learning, allowing representations to emerge naturally from the interaction between perceptual and learning systems.

1.5 Novel Contributions of This Thesis

We present a computational model that integrates CNN-based visual processing with associative learning mechanisms. This integration represents a significant advance in understanding how complex stimulus representations form through learning.

1.5.1 Integration of Visual Processing and Associative Learning

We demonstrate that integration between CNNs and associative learning is possible through a fully developed computational model. This hybrid system incorporates a biologically plausible image processing mechanism using convolution and pooling layers from a CNN. These layers extract hierarchical visual representations that serve as inputs to an associative learning model based on the DDA framework [10].

The model uses five convolutional layers, chosen to parallel the hierarchical processing stages in the ventral visual stream. Each layer progressively extracts more complex features, from simple edges in the early layers to complex shapes in deeper layers.

1.5.2 Elemental Framework with Emergent Configural Properties

The model uses an elemental approach where every data point of the visual representation is modelled as an element. When stimuli are presented, elements become activated, and shared elements across stimuli enable generalization and mediated learning. Despite this elemental foundation, the model successfully discriminates between elements and compound stimuli, as demonstrated in our negative patterning experiments.

The learning mechanism follows Hebbian principles [15], where element activity drives learning through an unsupervised error correction framework. The algorithm adjusts weights based on the distance between element activities, without requiring concepts of reward or punishment. The focus is on forming unitized representations through element-element associations.

1.5.3 State-of-the-Art Architecture for Representation Extraction

To our knowledge, this model is unique in its ability to extract and visualize the complex representation structures formed during associative learning. This connectionist architecture stores information in adaptive weights rather than symbols or rules, providing a more biologically plausible account of learning [16].

The model accounts for fundamental phenomena including acquisition, extinction [12, 17], blocking [18], conditioned inhibition, and negative patterning [19, 20, 21]. Importantly, it can extract and visualize the learned representations, providing direct evidence of how associative mechanisms shape perceptual representations.

The capability to visualize learned representations offers insight into the formation of complex stimulus structures during learning, making it a valuable tool for future research in both learning theory and computational modeling.

1.6 Thesis Structure

The remainder of this thesis is organized as follows:

Chapter 2: Literature Review – A comprehensive examination of associative learning theories from early S-R models through modern computational approaches, and an overview of CNN architectures relevant to biological vision.

Chapter 3: Methods – Detailed description of the research methodology, experimental design principles, and evaluation framework.

Chapter 4: Computational Model Design and Implementation – The complete model specification, including CNN architecture, associative learning mechanisms, and technical implementation details.

Chapter 5: Model Experiments and Results – Comprehensive results from simulations of classical conditioning phenomena including acquisition, extinction, blocking, conditioned inhibition, and negative patterning, with measures (V-values) and novel receptive field visualizations.

Chapter 6: Discussion and Future Work – Theoretical implications of the findings, model limitations, and directions for future research.

This work represents a significant step toward understanding how perceptual and associative processes interact to form complex representations, offering new insights into longstanding questions in learning theory while providing a practical computational framework for future research.

Chapter 2

Literature Review

Mechanisms of association that underlie cognitive function, specifically those that focus on knowledge acquisition, have been central areas of research among several scientific communities: Artificial Intelligence (AI), cognitive neuroscience, philosophy, psychology, economics, linguistics, and physics [22, 23, 24, 25, 26, 27]. Empirical studies in cognitive science have been in the limelight for decades with research-led initiatives that focus on connectionism [28, 29], which give association mechanisms a central place that accounts for behavioural findings and neurobiological correlates of human and animal learning [7]. These systems rely on parallel processing and try to incorporate functional properties of the brain, which are key for cognition [16]

2.0.1 Associative Learning

Associative learning is a fundamental theoretical approach that aims to understand the precise mechanisms by which humans and animals learn to pair events within their environment. Exposure to a contingency or correlation of events results in the formation of associative structures only when events are informative and are characterized by circumstances that produce learning [30]. The Neural links formed as a result of event pairing build different categories of Associative structures within stimuli and between different stimuli [31].

1. Animals learn about the existence and properties of stimulus S1 from a simple presentation of S1. The underlying assumption is that animals have an unconditional ability to detect the features of a given stimulus via the different sensory organs. Thus, neural centres (or nodes) representing S1 elements are activated simultaneously when these features are detected. Elemental nodes, are therefore said, encode activity information about these features. Whenever two or more nodes are activated concurrently, excitatory associations are formed or strengthened between these nodes. This associative learning framework follows the general Hebbian rule [15], and the associative links establish a super-ordinate structure constituting a mental representation of the entire stimulus [8]. It is the formation of such a unitised structure of representation that forms the theoretical basis for the development of our model. Learning of such representation structures relies on stimulus elements associations.

2. Animals also learn about the structure of the environment, which is modelled as relations among different stimuli. Animals learn associations between events, say S1 and S2, when sufficient information is given by one stimulus over the other [30]. Pavlovian conditioning is the mechanism by which a stimulus S2 comes to signal another stimulus S1. In this formulation, S1 can be an unconditioned stimulus (US) and S2 a conditioned Stimulus (CS). Exposure of both stimuli simultaneously or successively in the same context and close temporal

contiguity drive learning by concurrently activating representational nodes. Under these conditions, associative links are formed between the stimulus representational nodes. Activation of one node induces activation of the other representational node, an activation which is proportional to the strength of associations (or weight) between the links. However, mere contiguity fails to capture the relation required to produce an association, highlighting that for an association to be formed, one element must give relational information about the other [30]. We refer to this association as a stimulus to stimulus association (S-S). Building these unitised representations that follow an element-element learning approach is key to our learning model.

3. Animals also learn a given response, which results from experiencing events that relate S1 and the animal's own behaviour (S-R). This learning framework is often interpreted within the traditional framework of the reflex theory. An animal learns to perform a response (R) whenever a specific stimulus (S) is experienced. The assumed associative structure underlying this type of learning does not include a reward or a representation of a goal behaviour to be learned. This means that the animal performs the behaviour automatically, as a habitual response, when they see a given stimulus. A good example is the knee-jerk reflex. With repeated training, one learns how to kick as soon as the hammer is available. This type of learning constitutes the foundational theory of Reinforcement learning, in which the reward is used as a catalyst to learn the knee-jerking reflex in addition to other mechanisms, but a change in the value of the reward value

will not influence the probability of the learned habit response. What is learned by the animal is the relationship between the stimulus and the response. Discussions in Reinforcement Learning studies [32, 33] describe reward as a value that modifies the value of the link or acts as a catalyst to strengthen the link, not forming part of the S-R associative structure, which remains intact once it is formed and impervious to any change in the reward conditions.

4. Under associative learning theory, the mechanism by which animals learn to perform goal-directed behaviour, requires a representation of the outcome or goal. In this context, animals must learn about the relation between the consequences and their action. If, for instance, an animal who presses a lever (R) to get food pellets (O) - a rewarding outcome, learns to associate the lever press response with the delivery of food. As a result, the execution of the action results in a change in the environment of the animal leading to the formation of an associative structure in which a response is linked to an outcome (R-O) link. The reward (the goal) is thus part of the associative structure. If the value of the reward is changed, the value of R is altered. In addition, if the contingency is changed, for instance, if there is an automated way of getting the food pellets, the association would also be modified -there will be no need to press the lever.

Early theories of associative learning focused on S-R theories where learning was described as a substitution of a previously existing stimulus into an existing reflex system [34]. However, research conducted [35] posited that animals have mechanisms to adapt to relatively short-term variations that occur within the environment. They adjust to the 'causal texture' of their environments responding to the spatial, temporal and predictive relations among these events. The question that pervades modern conditioning is how much of this structure is represented by the organism.

Several of the approaches from the early years of learning theory research focused on experimental studies and establishing conditions necessary or sufficient for association formation and the elements that enter into associations. In parallel, experimenters' efforts were crowned by deploying quantitative models that account for association, which have since gradually grown in sophistication. These tools analyze learning at different levels, considering the spatial-temporal factors influencing learning.

A cardinal example of associative learning is Pavlovian Conditioning. Learning results from the exposure to relations among events in the environment, which enables animals to predict which consequences will follow each cue and adapt their behaviour accordingly. Central representations of elements responsive to environmental stimulation are linked so that activation of one stimulus can excite its associate [8]. Forming these connections among representations is referred to as associative learning. The underlying mechanism enables the animal to establish a causal structure of its world. Although early approaches in experimental psychology have long been infused with general 'ad hoc' rules that account for regularities in learning, these propositions provided insights that play an informative role in formulating more elaborated, specific, and accurate computational models. Modern conditioning envisions Pavlovian conditioning as a rich study area, which no longer focuses exclusively on the acquisition of conditioned responses, but it conveys causal and predictive information regarding the events of the environment, and, critically to this research, it accounts for how representations are formed and how animals.

Furthermore, with the modern advent of deep learning, associative learning computational models have the potential to be used to break down the learning process that underlies these deep artificial neural networks, which have remained a black box. In this chapter, we discuss the theories that have been developed and how they have evolved to account for Associative Learning.

2.0.1.1 Linear Operator Learning Tool

Early formal theories of associative learning [36, 37, 38, 39] interpreted habit as a process where stimulus (S) and Responses(R) become linked and constituted, and this association formed the cornerstone of learning. These theories aimed at explaining all learning phenomena and extending their postulate to cognition, in general, based on simple S-R connections. Although the effects of rewards were considered to be the result of obtaining a stimulus input (the US in Pavlovian conditioning), the latter was not considered part of the learning structure. Rewards were mere incentives able to motivate the production of a response, and as such not coded in the process of acquiring the habit. Among the most relevant theories at the time, it is worth highlighting Hull's and Estes' stochastic models [37, 4, 40]. These theories suggest that a response is evoked in the presence of a stimulus when the response reduces the drive. A drive is an arousal caused by biological or psychological needs that create an unpleasant internal state. According to Hull, a connection between an S and a paired R is formed when the R is followed by a reduction in this state of biological urge, reinstating the equilibrium (homeostasis) of an organism. Hull proposed a mathematical deductive theory of behaviour which assumes that the CS initializes a trace whose intensity changes over time. The rate of learning is proportional to the intensity of the CS trace at the time of reinforcement. When a habit is learned its strength changes during reinforcement according to this learning rule:

$$S_{Er} = (V * D * K * J * S_{Hr}) - S_{Ir} - Ir - S_{Or} - S_{Lr} \tag{2.1}$$

Where $\boldsymbol{S_{Er}}$ is the excitatory potential or the likelihood that an organism will produce a response $(\boldsymbol{R})$ to a stimulus $(\boldsymbol{S})$. $\boldsymbol{V}$ is the stimulus intensity dynamism that determines the influence of one stimulus over the others. $\boldsymbol{D}$ is the strength of drive determined by the amount of biological deprivation. $\boldsymbol{K}$ is the incentive motivation or the magnitude of the goal(amount of reward). $\boldsymbol{J}$ is the delay before the organism is allowed to seek reinforcement. $\boldsymbol{S_{Hr}}$ is the habit strength established in the previous conditioning. $\boldsymbol{S_{Ir}}$ is conditioned inhibition caused by a previous lack of reinforcement. $\boldsymbol{Ir}$ is reactive inhibition, and $\boldsymbol{S_{Or}}$ is a random error value while $\boldsymbol{S_{ELr}}$ is reaction threshold, or the smallest amount of reinforcement that will produce learning.

If we defined habit as the likelihood that an organism will produce a response $(\boldsymbol{R})$ to a stimulus $(\boldsymbol{S})$, then the acquisition curve of this conditional response in Hall's formulation could be understood as a form of delta error for habits [6]. This can be expressed as follows.

$$\boldsymbol{\Delta H_i = \theta(M - H_i)} \tag{2.2}$$

Where the change in habit, $\boldsymbol{\Delta H_i}$, is the discrepancy between a maximum value, $\boldsymbol{M}$ and the current strength of habit $\boldsymbol{H_i}$. $\boldsymbol{\theta}$ is the learning rate. The strength of the habit, according to Equation 1, follows a negatively accelerated relationship with the number of reinforced trials.

Estes stochastic learning [41, 40, 4] is also a form of S-R learning paradigm. In a conditioning trial, a randomly selected and partial subset of elements is sampled to form an association with the response. Reinforcement is defined as an experimental condition that ensures a successive occurrence of the response that will be contiguous with a new sample of elements from the same stimulus population. $(\boldsymbol{S - R})$ contiguity determines the association, which is mediated by reward; hence, $\boldsymbol{S}$ elements in the sample are associated with the response as a consequence. In subsequent trials, new samples are drawn, and new elements are associated with the response. Conditioning is complete only when most of the population is conditioned. Estes [4] formalized this learning by:

$$\boldsymbol{\Delta p_i = \theta(\pi - p_i)} \tag{2.3}$$

Where $\boldsymbol{\pi.}$ is the maximum proportion of elements that are conditionable, and $\boldsymbol{p_i}$ is the proportion already conditioned while $\boldsymbol{\theta}$ represents the sample size. This theory stipulates that the CR is a function of the probability of a response in the presence of a stimulus $\boldsymbol{S}$.

Although these two approaches hold different concepts about Pavlovian conditioning, they were hardly consistent in providing a comprehensive explanation of what is today considered a variety of different learning mechanisms. Initial quantitative approaches accounting for acquisition [42, 43] defined abstract concepts such as associative strength quantified by observable variables such as number of trials or frequency of conditioned responses.[6] However, what can be deducted is the need for an error correction framework in the underlying associative mechanisms. The change in the strength of the links is majorly determined by the proportion of the difference in the sampled elements and those that have been conditioned, as well as the difference in maximum strength in a habit and current habit strength of the animal.

2.0.1.2 Law of Effect

Thorndike [36] interpreted conditioning as the strengthening of new responses by their consequences. He elaborated this through his original proposal, the 'law of effect' where cats learned to press a catch or pull a loop of string to escape from the puzzle-box. The response $\boldsymbol{R}$ was only associated with a given stimulus$\boldsymbol{S}$ after a satisfying state of affairs followed it. The 'law of effect' formed the cornerstone of instrumental learning [38], where a given goal-directed behaviour is determined by past contingencies of reinforcement. Hull's theory, on the other hand, stated that $(\boldsymbol{S}-\boldsymbol{R})$ links are strengthened only when a reduction in drive occurs. Pavlov's [44] explanation of the formation of new habits, however, differed from Thorndike's and Hull's explanations. Conditioning was conceived as establishing new units of behaviour, conditional or conditioned reflexes. These units comprised a conditioned stimulus $(\boldsymbol{CS})$ and a response also known as conditioned response $(\boldsymbol{CR})$. The establishment of the $(\boldsymbol{CR})$ was assumed to be dependent on reinforcement such that an initially neutral stimulus $(\boldsymbol{CS})$ repeatedly paired with an unconditional stimulus $(\boldsymbol{US})$ that elicits a response, Unconditioned response $(\boldsymbol{UR})$. In this case, conditioning was a matter of the $(\boldsymbol{CS})$ substituting the $(\boldsymbol{US})$ and thus eliciting a response $(\boldsymbol{CR})$. If two stimuli are presented in close temporal contiguity and the $\boldsymbol{US}$elicits some reaction, then the $\boldsymbol{CS}$ will evoke a similar response.

2.0.1.3 Beyond Contiguity: Cue Informativeness

Hull's law of reinforcement, Estes' sampling theory, Thorndike principle of law of effect, and Pavlov's substitution theory assume a smooth learning gradient for all stimuli involved in training. The way a representation of stimulus was presented in early onset theories, however, has been challenged by a series of experiments conducted with compound cues such that was learned by one stimulus appeared to depend on the associative value of the other stimulus [45, 46, 47, 48, 49, 17, 50] A good example that can describe this phenomenon is Blocking [47, 18]. Subjects are conditioned using stimulus $\boldsymbol{A+}$ prior to receiving compound training of $\boldsymbol{AB+}$. Upon testing, the novel cue $\boldsymbol{B}$acquires little or next to no conditioning. It is extrapolated that learning depends on the surprisingness of the $\boldsymbol{US}$, the associative link formed by $\boldsymbol{A}$ to the outcome prevents the formation of an equivalent link between $\boldsymbol{B}$ and the $\boldsymbol{US}$. As a result, $\boldsymbol{B}$ is not a strong predictor of the $\boldsymbol{US}$. This experiment was pivotal as it sheds light on results discovered by Pavlov [51] on overshadowing where a compound stimulus acquires less associative strength than when trained separately [18]. Additionally, when two different compounds, $\boldsymbol{AX}$ and $\boldsymbol{BX}$, are equally reinforced $(+)$ and non-reinforced $(-)$ 50% of the time (i.e $\boldsymbol{AX}+/-$, $\boldsymbol{BX}+/-$) the absolute predictive value of the common features is higher than when that stimulus compounds more reliably signal the occurrence and nonoccurrence of reinforcement ($\boldsymbol{AB}+$, $\boldsymbol{AB}-$) [52]. In other words, although X received an identical amount of reward in both conditions, its predictive value differs depending on the values of their partners. Additionally, if the $\boldsymbol{CS}-\boldsymbol{US}$ contiguity is preserved but the frequency of the US is varied in the absence of the $\boldsymbol{CS}$, conditioning is high when the probability of the $\boldsymbol{US}$ in the absence is low [49].

The notion of contiguity and reinforcement have been considered crucial factors to learning [6]. Concurrent activation of the stimuli representational nodes is a determinant of learning, but not by itself. Cue competition phenomena have shown that the co-activation is not a sufficient condition. The total amount of available strength for a given connection is determined by the weight of all other active nodes linking to the same outcome. Such models have been developed to account for the variation of interaction in either the $\boldsymbol{CS}$ or the $\boldsymbol{US}$. Three categories have been identified: First, the model assumes that the $\boldsymbol{CS}$compete for the reinforcement, it assumes that there is competition among the $\boldsymbol{CS}$ for attention, and lastly, there is a model that appeals to both processes.

2.0.1.4 Rescorla Wagner Model: variation in US processing

The Rescorla-Wagner$(\boldsymbol{RW})$ model assumes that learning involved in a $\boldsymbol{CS}$ depends on the current associative strength held by all cues present in that trial [11, 12]. Any change in associative strength of a given $\boldsymbol{CS_i}$ is given by:

$$\boldsymbol{\Delta V_i = \alpha_i \beta_j (z_j \lambda_j - \sum x_i V_i)} \tag{2.4}$$

where $\boldsymbol{\alpha_i}$ and $\boldsymbol{\beta_j}$ are learning rates determined by intensities of the $\boldsymbol{CS_i}$ and $\boldsymbol{US_j}$. The values of $\boldsymbol{x_i}$ and $\boldsymbol{z_j}$ are 1 when the corresponding stimulus is present and 0 when absent. $\boldsymbol{\sum x_i V_i}$ represents the sum of all the associative strength of all other stimuli present in the trial at that point in time. *Excitatory learning* takes place when the difference between activation of the adaptive unit by the $\boldsymbol{US}$ is greater than activation by all the $\boldsymbol{CS}$ present in that trial while *Inhibitory learning* is experienced when the activation of the adaptive unit by the US is less than the activation by all the CS's that are present in the trial. The main difference between $(\boldsymbol{RW})$ and the linear operator rule is the incorporation of the aggregate associative strength in the computation of change in associative strength resulting in a 'global' prediction error term. This rule is compared to a supervised learning problem [53] where the activation by the $\boldsymbol{US}$is regarded as the teacher, whereas the activation by all the $\boldsymbol{CS}$ present in the trial is regarded as the predicted output. The discrepancy between the two outputs, real and predicted, produces an error that is reduced by a margin for every trial. Reinforcement is maximum when the discrepancy is maximum but reduces as the CS acquires the strength to activate the adaptive unit. While this model has several limitations, it plays a fundamental role in understanding how Pavlovian conditioning may contribute to learning stimuli representations.

2.0.1.5 The attentional Model of Mackintosh: variation in CS processing

The attentional theory assumes that animals attend to cues that are relatively better predictors of outcome than others [48]. Attention to a cue is modelled as a change in proportion to the relative predictiveness of the cue for the outcome. Associative failure is caused by the diminution of attention paid to the target CS, also referred to as CS processing failure [6]. Stimuli compete for a limited attentional capacity such that when a given CS is preferred over another, selective attention is learned and retained for future learning which presumably aids in reducing proactive interference between stimuli, thereby speeding up learning. RW-model is unable to account for the pre-exposure effect because there is no US, as the error term remains zero. A pre-exposed CS retards the subsequent acquisition of both excitatory and inhibitory learning [54, 55]

Mackintosh [48] postulated that the associability, α, of a CS increases when a given CS is the best predictor of the US relative to all other cues present. A formal description of the associability, α_i, of a CS_i increases according to **Equation 5** and decreases as per **Equation 6**

$$|\boldsymbol{z\lambda} - \boldsymbol{x_i V_i}| < |\boldsymbol{z\lambda} - \boldsymbol{x_j V_j}| \tag{2.5}$$

$$|\boldsymbol{z\lambda} - \boldsymbol{x_i V_i}| \geq |\boldsymbol{z\lambda} - \boldsymbol{x_j V_j}| \tag{2.6}$$

Where $_j$ represents all other CS's present in the trial. The proportion of increase or decrease of a CS's associability, $\boldsymbol{\alpha_i}$, is given by the difference as depicted in equation 7

$$|\boldsymbol{z\lambda} - \boldsymbol{x_i V_i}| - |\boldsymbol{z\lambda} - \boldsymbol{x_j V_j}| \tag{2.7}$$

Mackintosh [48] uses Hull's principle [56] of non-competitive $\boldsymbol{US}$ error rule to compute the associative change. The $\boldsymbol{US}$ has a limited role in explaining the acquisition of associative strength of a given stimulus. Mackintosh model explains unblocking where the surprising partial omission of reinforcement during blocking attenuates the blocking of a cue [57]. Thus, it avoids the prediction of over-prediction pushing the blocked cue toward becoming inhibitory, which the RW model predicts in some circumstances for this treatment. This model also accounts for learned irrelevance [58, 59] whereby a CS uncorrelated with US presentations shows poorer subsequent acquisition because of the best relative predictor accruing the most associability, and hence conditioning more quickly than competing cues.

The presence of a $\boldsymbol{US}$ results in an increment of an excitatory tendency towards an asymptote of, say +1, whereas the omission of an expected US results in the increment of an inhibitory tendency towards, say -1. The net associative strength of a CS, V, is the difference between excitatory and inhibitory associations with the US [48]. The model, however, is not computationally viable; there are no real values for the change in associability, only tendencies. Some remarkable difference between the Mackintosh model and the RW model lies in the concept of inhibition. In the RW model, non-reinforced presentations of an inhibitory CS result in the suppression of a response, and the the associative strength of a cue gravitates towards 0. On the contrary, in the Mackintosh model, inhibitory properties push associative values towards -1, opening a broad spectrum of reinforcing effect variations. Yet, this very same reason prevents the Mackintosh model from explaining super-conditioning. Mackintosh model has been improved to address the challenges of super-conditioning wherein presenting a conditioned inhibitor of an outcome together with a novel cue leads to stronger excitatory conditioning of the novel cue than if it were conditioned individually. The extension, made by LePelly [60], can account for a computational version that can be compared with RW model from a computational perspective.

2.0.1.6 The Pearce and Hall mixed model: variation in both CS and US processing

Pearce and Hall [61] proposed a model that relied on both CS and US processing. According to this model, unexpected outcomes are more liable to attention and hence win more associative strength. the Pearce and Hall model [61] can successfully predict phenomena like the Hall and Pearce negative transfer effect in which acquisition of a conditional response is retarded when the CS is previously paired with the same US but of a lower intensity [62]. Following the postulates of the model, the CS should lose associability during the pre-training phase, which would put it at an attentional disadvantage in comparison to a novel stimulus, thus slowing down the conditioning rate. This model, however, cannot explain why a CS that was previously paired with the same US retards when paired with the same US but of a lower intensity [48].

According to the Pearce and Hall model [61], the associability of a $\boldsymbol{CS_i}$, $\boldsymbol{\alpha}$ decreases when it is followed by a consistent outcome [61]. However, when the outcome is inconsistently predicted, i.e., when the expectancy of reinforcement is not confirmed, the associability of the stimulus is sustained. In other words, the animal has a limited capacity to process both the CS and the US simultaneously, and only unexpected events get access to the processor. The associability of a $\boldsymbol{CS_i}$ on a trial is proportional to the difference between the activation of the adaptive unit by the US and its activation by all the CS's in a previous trial $\boldsymbol{n-1}$. Therefore the model easily accounts for the negative transfer effect. On the other hand, the model is unable to account for learned irrelevance, as the predictor in a learned irrelevance procedure will accrue more associability because of their lack of correlation with the occurrence of an outcome. In the context of a standard acquisition and extinction protocol, the PH model predicts a sudden increase in associability when the contingency is changed with a gradual decline thereafter [10]. The total net prediction is defined as the difference between excitatory and inhibitory predictions [63] this is expressed as:

$$\boldsymbol{\alpha}_i^n = \gamma|\boldsymbol{z\lambda}^{n-1} - (\sum \boldsymbol{x}_i \boldsymbol{V}_i^{n-1} - \sum \boldsymbol{x}_i \overline{\boldsymbol{V}_i^{n-1}})| + (\mathbf{1}-\gamma)\boldsymbol{\alpha}_{\mathbf{1}}^{n-1} \quad (2.8)$$

The excitatory $\sum \boldsymbol{x}_i \boldsymbol{V}_i^{n-1}$ and inhibitory $\sum \boldsymbol{x}_i \overline{\boldsymbol{V}_i^{n-1}}$ learning develop separately for a given CS. γ governs the relative influence of the preceding trials. For $\gamma \approx 1$, the associability will be determined by the events of the immediately preceding trial n; while for $\gamma \approx 0$, $\boldsymbol{\alpha}$ is determined almost exclusively by earlier trials. Excitatory learning accruing to a given CS$_i$ in a given trial n is proportional to the it's associability $\boldsymbol{\alpha}_i^n$, Salience $\boldsymbol{S}_i$ and the activation of the adaptive unit by the US, $\boldsymbol{z\lambda}$

$$\Delta \boldsymbol{V}_i^+ = \boldsymbol{\alpha}_i^n \boldsymbol{S}_i \boldsymbol{z\lambda} \quad (2.9)$$

Inhibitory learning accrued to a CS$_i$ in a given trial n is proportional to the associability of a CS $\boldsymbol{\alpha}_i^n$, Salience $\boldsymbol{S}_i$ and the discrepancy between the expected activation of the adaptive unit by the all the CS $\sum \boldsymbol{x}_i \boldsymbol{V}_i$and the Activation of the adoptive unit by the US $\boldsymbol{z\lambda}$.

$$\Delta \boldsymbol{V}_i^- = \boldsymbol{\alpha}_i^n \boldsymbol{S}_i (\sum \boldsymbol{x}_i V_i - \boldsymbol{z\lambda}) \quad (2.10)$$

The net associative strength of a give CS$_i$ is given by subtracting the inhibitory strength from the excitatory associative strength.

2.0.1.7 Elemental and Configural Representational Approaches

The simulation pattern for the Associative mechanism in RW model [11], Mackintosh [48], and Pearce and Hall [61] is elemental. This means that conditioning comprises separable elements that develop associations with the $\boldsymbol{US}$. This view was challenged when a decrement in $\boldsymbol{CR}$ was observed in a well-trained $\boldsymbol{CS}$ when presented as a compound with a novel untrained stimulus [13, 14, 64], also termed as external inhibition. Models that are elemental cannot account for non-linear discrimination, as they assume the response accrued to a compound is proportional to the sum of the associative strength of the elements. John Pearce [13, 14] introduced the Configural view of learning where elements of stimulus $\boldsymbol{A}$ and compounds $\boldsymbol{AB}$ have different configurations. An element $\boldsymbol{A}$, and a compound $\boldsymbol{AB}$, have two different representations. Two benchmarks of non-linear discriminative performance of a model have been negative patterning ($\boldsymbol{A}+$, $\boldsymbol{B}+$, $\boldsymbol{AB}-$) and bi-conditional ($\boldsymbol{AB}+$,$\boldsymbol{CD}+$, $\boldsymbol{AC}-$, $\boldsymbol{BD}-$) discriminations. Elemental models cannot break down the linearity of the compound trials. The animals must learn to withhold responding on the trials with two cues which individually predict a common outcome. Bi-conditional discriminations involve more complex non-linearity. Simple summation of individual cues offers no information for solving the discrimination[10].

These configurations are the basic functional units in a conditioning situation [14, 64]. The associative strength of the configuration $\boldsymbol{C}$, $\boldsymbol{V_C}$, is given by the sum of the associative strength directly conditioned to the configuration, $\boldsymbol{E_C}$ and the associative strength that generalizes to the configuration from other configurations $\boldsymbol{e_C}$. The value $\boldsymbol{e_C}$ is determined by the direct strength of those configurations weighted by the similarity index to the target configuration, C, where the index is determined by the number of similar elements shared. Formally $\boldsymbol{V_C}$ is given by:

$$\boldsymbol{V_C = E_{Ci} + \sum S_{Ci} E_i} \qquad (2.11)$$

Where $\boldsymbol{E_i}$ is the direct associative strength of any configuration $\boldsymbol{i}$ and $\boldsymbol{S_{Ci}C}$ is the similarity index between configurations $\boldsymbol{C}$ and $\boldsymbol{i}$. The change in the direct associative strength of the configuration in a given trial is expressed as:

$$\boldsymbol{\Delta E_C = \alpha_C \beta (z\lambda - x_C V_C)} \qquad (2.12)$$

Pearce's model differs from the RW model based on the assumptions made about representations. First, a compound is represented as a single sensory unit in the Pearce model in contrast to the RW model. Additionally, for the Pearce model, the notion of associability does not vary with the number of nominal stimuli included in the configuration; similarity salience, $\boldsymbol{\alpha}$, is assigned to each configuration despite the number of components. Whereas the RW model assumes that a compound has a greater salience than any of its components approximated by the sum of the individual component salience $\boldsymbol{\alpha}$. These assumptions lead to different results between the two models hence varying the results of generalization and discrimination learning [6].

An attempt to translate the Pearce model to an elemental model for easy comparison to an RW model was made [65]. It was assumed that when stimuli A and B are compounded, some elements are inhibited [66]. This version of the Pearce Model makes an assumption that salience is equal for all configurations, and theoretical elements in a compound remain invariant. Additionally, there is a level of statistical independence when Stimulus A is compounded with another stimulus B. The subset of elements active when A is active is independent when A is compounded with C. The main difference is that the Pearce model forms compounds by inhibiting some elements, whereas the RW model forms compounds by adding some elements [66].

An experiment conducted involving rabbit eyelid conditioning presented experimental evidence that could not be accounted for by either the Pearce model or the RW model. The training involved a conditioning procedure with stimuli A, AB and ABC and testing with the same cues. Whenever a stimulus was added, i.e., training with A and testing with AB and ABC, as well as training with AB and testing with ABC, the was a decrement in response with every addition of a stimulus. This confirmed the external inhibition effect favouring the configural inhibited elements approach over the elementalistic addition approach. Moreover, it was noted that there was even a greater decrement when removing a cue from AB to A compared to adding a cue from A to AB. The results contradicted the Pearce model, which predicates that the decrements should be symmetrical [6].

A model [64, 66, 65] that accounts for the asymmetrical decrements was proposed [66]. This model, *Replaced units Model* (REM), is a computational concept of the hypothesis of the afferent neural interaction hypothesis [67]. New configural elements are activated when stimuli are presented as a compound, and additional elements are inhibited. Some elements activated by the individual stimuli are replaced by the addition of unique configural elements activated by the stimulus compounded, and some are removed (unique elements) when the cue is presented in a compound. The added elements correspond to elements representing context-dependent features of the stimulus, while the replaced elements are assumed to represent features of the cue that are uniquely present when the cue is presented alone. This model accounts for negative patterning and bi-conditional discrimination in the same way as the RW model because the representation of a compound is distinct from its constituent elements. It also accounts for the difference in responding between single and compound stimuli. The downside is that in an elemental model, RW and REM assume that a redundant cue will facilitate learning, but that is not observed. Additionally, it is assumed that complete reversibility of an association between a stimulus and the outcome will be successful should the previously learned contingency be reversed. However, experiments display retroactive interference in feature-negative discriminations for both RW and REM models.

2.0.1.8 Simple Real-Time models

Timing effects among stimuli play a role in the formation of associations and the generation of responses. There are different timing models that account for cognitive processes in time. There are also associative learning timing models that account for time factors of the stimulus processes, and lastly, there are models that fall in between Associative learning timing models and cognitive timing models. We are only going to consider Associative learning timing models that are not in real-time, but they implement the discretized temporal factors that usually would not be accounted for by the models that have already been reviewed.

Temporal factors such as the time interval between the onset of a CS and US, also known as inter-stimulus Interval (ISI), majorly affect the latency to initiate CR and the peak of the CR. It has therefore been a major factor to consider while formulating real-time models [68, 69, 70, 71, 72]. Real time here refers to discretized models. To account for these findings, early models suggested that CR varied in proportion to the strength of conditioning. Poor learning was manifested in the form of slow-rate of acquisition, low CR amplitude and a longer latency of initiation and peak. Additionally, the ISI function hypothesis [73, 74, 75, 76] posits that the ISI determines the rate of learning. This theory suggests that an intermediate ISI is an optimal index where learning is maximal and that acquisition decreases exponentially with longer and shorter ISI's

One key temporal feature that affects learning and stimulus representation in these simple Real-time models is the concept of trace [44, 56], which refers to the change of stimulus representation over time. Associations are formed to each part of the trace depending on the presence or absence of reinforcement and strength of the trace. This was built according to

Hull's [56] S-R principle that had a delay component in its formulation. Associations are, therefore, formed differentially across time. The activity of each processing unit reflects a time-varying course of activity in the stimulus during the trial.

2.0.1.9 Real-time Rescorla Wagner Model

This model [77] [78] is an extension of the RW model [11, 12] that accounts for within-trial associations. The CS trace $x(t)$ follows a curvilinear function increasing in a negatively accelerated and decays to zero after CS offset [77]. The US trace $z(t)$ is a rectilinear binary function that assumes a value of 1 when the US is active and 0 when it is inactive. Brandon [78], on the other hand, processes traces in a momentary fashion by use of time-steps. Both stimuli, US and CS, traces are characterized by an initial period of negatively accelerated rise of activity followed by a period of adaptation then they decay back to inactivity. Learning involves computing changes in associative strength at each moment of time by taking into account factors such as the momentary strength of the CS trace and the reinforcement.

Simple real-time implementation of the RW rule consists of computing changes in associative strength for a given CS at each moment, according to the momentary strength of the CS trace and the reinforcement. This notion has been formalized in the following equation:

$$\Delta V_i = \alpha_i \beta_j [z_{(t)} \lambda_{(t)} - \sum x_{i(t)} V_{i(t)}] x_{i(t)} \tag{2.13}$$

The amount of learning accrued to a CS$_i$ at any moment in time (t) is given by the discrepancy in time between the activation of the adaptive unit by all the CSs available in that trial and the US i.e $z_{(t)}\lambda_{(t)} - \sum x_{i(t)} V_{i(t)}$ controlled by the strength of the trace of CS$_i$ given by $x_{i(t)}$. The net associative strength is given by an overlap of inhibitory and excitatory tendencies evoked by the presentation of the CS at every moment in time.

Brandon's [78] trace definition posited a greater acquisition with an ISI of intermediate duration, which was similar to what was reported by Schmajuk [77]. When trained with many trials, the function relating the ISI and associative strength shifts from an inverted U-function to a linear decreasing function, which means that conditioning is best for shorter ISIs.

2.0.1.10 The time derivative Model of Richard Sutton and Andrew Barto

Time derivative models propose that both CS and US have reinforcing properties. The adaptive unit is not required to play the role of a supervisor to produce changes in the Associative strength of a CS. The model accounts for second-order conditioning, where the acquisition of the associative strength by a CS$_i$ is with another stimulus CS$_j$ which had previously been conditioned with the US. Therefore reinforcement is based on the difference in the output between two time intervals such that a CS with associative strength greater than zero can produce reinforcement in the absence of a US. The first time derivative model to be proposed was called the Sutton and Barto (SB) model [79]

In this model, the CS generates two traces, Stimulating trace $x_i(t)$ and eligibility trace, $e_i(t)$. $x_i(t)$ is a binary function that generates responses, which takes a value of 1 when the CS is active and 0 when inactive. $e_i(t)$ continually changes with calculations of a running average of $x_i(t)$ and decays with the offset of the CS. The US representation is similar to the stimulating trace of the CS. The change in eligibility trace is given by:

$$\Delta e_i(t) = \delta(x_i(t) - e_i(t)) \tag{2.14}$$

Where δ is the rate of change of $e_i(t)$. Since the stimulating trace is a binary function throughout the duration of the CS, $e_i(t)$ continually increases towards 1 at the onset of the CS at a fixed rate, δ, and decays at the same rate upon offset. Learning at any moment in time is given by the discrepancy between the current and the immediate previous output. Reinforcement is a function of the time difference between the output at time t and time $t - 1$. Therefore, the change in associative strength is expressed as:

$$\begin{aligned} \Delta V_i &= [Y(t) - Y(t-1)]e_i(t) \\ \Delta V_i &= \begin{gathered} \delta[(\sum x_i(t)V_i(t) + z(t)\lambda(t)) - (\sum x_i(t-1)V_i(t-1)+ \\ z(t-1)\lambda(t-1))]e_i(t) \end{gathered} \end{aligned} \tag{2.15}$$

The onset of the stimulus is assumed to generate excitatory associations, whereas the offset generates inhibitory associations. Trial-level phenomena such as blocking, overshadowing, and conditioned inhibition are accounted for in the SB model[79], but there are several predictions that don't match with empirical evidence [6]. SB model predicts that the level of conditioning in experiments with longer ISIs will be the same. This contradicts the evidence showing that there is poor conditioning when longer ISIs are used. This has been attributed to the fixed rate, δ, of change in the eligibility trace that increases towards the value of 1 and decays at a constant rate. Differential condition is only noticed in instances of Short ISI because the eligibility trace will not have reached the asymptote at the moment of US presentation. Another disadvantage is the inhibitory learning caused by the offset of the US causing negative reinforcement at a time, t producing lower output at time $t - 1$, i.e., $Y(t)< Y(t - 1)$. The eligibility trace of the CS, in most cases, overlaps with the US offset, resulting in inhibitory learning that reverses or cancels excitatory learning due to the US onset.

Improvements to this model were made to overcome the eligibility challenges. The drive reinforcement model (DR model) [80, 81] postulate that all CSs have the same eligibility trace independent of their duration, i.e., the CS is only incremented at its onset followed by a gradual decrement. Eligibility trace does not depend entirely on the time course of the CS but follows a fixed course after the CS onset independent of its offset resulting in long and short CSs generating exactly the same trace. This means that shorter ISIs will predict stronger conditioning.

Additionally, an SBD model [82] was developed to address the issue of long ISIs. This model posits that the rate of decay of the eligibility trace increases as a function of the CS duration. The eligibility trace decays after the CS offset, but the rate of decay is proportional to the duration of the CS.

This model proposes that the trace of the US decays progressively upon the offset of the u,s which reduces the inhibitory effects of the US observed in the SB model. Moreover, the reinforcing effects of the US is a function of the current associative strength, $\boldsymbol{V}$, which is similar to the RW model that implements a diminishing value of its respective US as a function of the current Associative strength. Although both the DR and SBD models solve the problem of long ISI delay conditioning, there is still a challenge of incorrect prediction of strong inhibitory simultaneous conditioning.

2.0.1.11 SOP

Stimuli representation in the SOP model [83, 84] is composed of a pair of processing units, primary and secondary, that are made up of a number of elements. The primary unit is associated with its respective secondary unit. Presentation of a stimulus invokes activation of some proportion of elements in the Primary unit, $\boldsymbol{A1_i}$, followed by activation of secondary elements $\boldsymbol{A2_i}$, recurrently inhibiting corresponding elements in $\boldsymbol{A1_i}$. The activity of each stimulus across time is represented by two traces, the trace of the primary and secondary unit. The rate of activation of each unit, $\boldsymbol{p1}$ and $\boldsymbol{pd1}$, primary and secondary, respectively, is independent of the presence or absence of the stimulus.

The US sensory and adaptive units have separate activities because the CR does not exactly mimic the UR. This model postulates that when the US is presented, two sequences of response are generated. The CS activates the US secondary unit through the associative links established and indirectly inhibits the primary unit through the secondary unit's inhibitory influence over the primary unit.

Two rules were proposed that relate the activity in the different units and generate responses [83]. One is the retrieval rule that states that CS influences the activity of the US secondary unit according to the product of the momentary proportion of its active primary (pA1i) and secondary elements (pA2i) and the net associative value (Vi)

$$\boldsymbol{P2US = \sum V_i(r1PA1_i + r2PA2_i), for, 0 < P2 < 1} \tag{2.16}$$

The products are summed to determine the conditioned activation of the US Secondary unit. $\boldsymbol{r1}$ and $\boldsymbol{r2}$ represent the relative weight of the respective primary and secondary units. Since p2US is restricted to the unit interval or ($\boldsymbol{0 < p2 < 1}$), a CS with the net inhibitory association has no effect on the activity of the secondary node, but it can contribute to making its activation more difficult in the presence of other excitatory cues [6]. The second rule was a response generated rule that states the response generated by units involving a mapping function $\boldsymbol{f_U S}$ weighted by linear factors $\boldsymbol{w1}$ and $\boldsymbol{w2}$ [83]. Both CS and US influence the generation of a response. The response R is expressed as

$$\boldsymbol{R = f_{US}(w1PA1_{US}1_i + w2PA2_{US})} \tag{2.17}$$

Excitatory and inhibitory links are established separately for both the CS and the US, and changes in excitatory links are proportional to the momentary product of active CS and US primary elements. Similarly, changes in inhibitory links at any moment are proportion to the product of CS primary elements and the US elements that are active. We can express this as:

$$\Delta V_i{}^{+}(t) = L^{+} \sum (PA1_i(t)x * PA1_{US}(t)) \tag{2.18}$$

$$\Delta V_i{}^{-}(t) = L^{-} \sum (PA1_i(t)x * PA2_{US}(t)) \tag{2.19}$$

$$\Delta v_i{}^{+}(t) = \Delta V_i{}^{+}(t) - \Delta V_i{}^{-}(t) \tag{2.20}$$

Where L^{+} and L^{-} are learning rates. The activity of the secondary US unit is influenced by all the CSs with active primary elements and V other than zero. Consequently, the activation of the secondary unit of the US is equivalent to the total prediction of the US or $\sum V$. The SOP model accounts for the priming phenomena as the acquired CS-US association not only influences the probability of response but also US processing. This has proven to be advantageous as the model deals well with challenges related to inhibitory learning that have proved to be difficult for models like RW to account for. For instance, according to the RW model, non-reinforced presentation of inhibitory CS results in the extinction of inhibitory properties but for SOP, inhibitory CS has no activity on secondary US units and therefore, no learning takes place. Another advantage of the SOP model is the precision associated with ISI - effects for a given number of trials. Unfortunately, with a high degree of training, the shape of the ISI function changes to a linear rather than an inverted U. This seems to be a prediction of any model in which inhibitory learning where opportunities for non-reinforcement become increasingly more influential with longer ISIs. This has consequently resulted in defining a stable-ISI function leading to a fully connected and real-time model.

2.0.1.12 SSCC TD: A Serial and Simultaneous Configural-Cue Compound Stimuli Representation for Temporal Difference Learning

The SSCC TD model [85] is an extension of the TD model. This model incorporates configural representations that are also required to instantiate key learning paradigms that rely on configural cues, such as discrimination and summation tests for inhibition. The notion of the configural cue is considered an emergent perceptual cue that represents a combination of given elements. This representation competes with other cues to gain associate strength like any other orthodox stimulus [86, 12].

Configural representations in an SSCC TD are built by the co-occurrence of stimulus simultaneously within a conditioning trial such that the respective activation of representations takes place at the same time. Formation of configural cue can also be between an active stimulus representation overlapping with memory traces of earlier stimulus. Additionally, configural compounds can be formed between context and the stimulus representation.

The model learns through error correction while incorporating compound stimulus configurations in a real-time architecture. This has been a successful model that explains the performance of learning tasks involving compound stimulus for which factors such as generalization and discrimination are inherent [85]

2.0.1.13 Hidden - Units Model

Most hidden-unit models were developed to account for discrimination problems that have challenged linear models for a long time. Negative Patterning, for instance, is one of the discrimination problems that have been problematic. CS are reinforced when presented separately but non-reinforced when they are presented as a compound (A+, B+, AB-). The typical result of this procedure is that the response of the animals to individual CS is more significant than their response to the compound. This is an example of an XOR problem, suggesting that the representation of a compound CS should be different from the representation of its components. Hidden-unit models suggest that a configural representation is encoded by the hidden units connected to the sensory layer and to the output layer. The SD model is an example of the hidden unit model that was proposed by Schmajuk and Dicarlo [87] and later improved [88, 77, 89].

This three-layered network has input units, hidden units, and an output layer. All the links are modifiable except for the US-to-output connection. This model was proposed as an improvement of the Pearce and Hall[61] model to account for negative patterning and address issues related to occasion setting [88, 77, 89]. After a short delay, the onset of the CS initiates a trace, $x_i(t)$, that increases to a maximum over time and decays to zero gradually upon the CS offset. The US is a binary function that assumes a value of 1 when the US is on and 0 when off. When activated, the hidden units initiate a trace of w_i that is determined by the weighted activity of the sensory unit, that is: $w_j(t) = \sum x_i(t)C_{ij}(t)$.

Associative change takes place in three sites: the connection between the sensory units and the output unit VS_i, the connections between the hidden units and the output units, VH_i, and in the connection between the sensory units and the hidden units, C_{ij}. The learning rule applied to VS_i and VH_i determines the generation of CR and RW learning rule applies. This is given by the discrepancy of the momentary activation of the output unit $z(t)\lambda(t)$ and the aggregated associative strength held by the active sensory unit and hidden unit:

$$\Delta VS_i = \theta_1 x_i[z(t)\lambda(t) - (\sum x_i(t)VS_i(t) + \sum w_j(t)VH_j(t))](|1 - VS_i|) \tag{2.21}$$

$$\Delta VH_j = \theta_2 W_i[z(t)\lambda(t) - (\sum x_i(t)VS_i(t) + \sum w_j(t)VH_j(t))](|1 - VH_j|) \tag{2.22}$$

where factors $|1 - VS_i|$ and $|1 - VH_j|$ maintain the associative values between -1 and 1 and θ_1 and θ_2 are learning parameters. The activity of the output unit requires that activation produced by sensory units and hidden units be differentiated. Thus $(0 < \theta_1 < \theta_2 < 1)$. This adjustments confers a competitive advantages to the configural (hidden) associations over the direct associations. Prediction error or output Error (EO) is given by $z(t)\lambda(t) - [\sum x_i(t)VS_i(t) + \sum w_j(t)VH_j(t)]$

Changes in the association between the sensory units and the hidden units are governed by the backpropagation learning rule. Error term for sensory unit-to-hidden units associations ($\boldsymbol{EH_{ij}}$) [29, 87]. This error is a function of the degree of activation of the adaptive unit by the hidden unit ($\boldsymbol{w_i(t)VH_i(t)}$) and output Error (EO):

$$\boldsymbol{EH = f((EO)VH_i \sum x_i(t)C_{ij}(t))} \tag{2.23}$$

Where $\boldsymbol{f}$ is a sigmoid function. The change in the association between the sensory unit, $\boldsymbol{S_j}$, and the hidden unit, $\boldsymbol{H_j}$, is given by:

$$\boldsymbol{\Delta C_i j = \theta_3 x_i EH(|1 - C_{ij}|)} \tag{2.24}$$

The learning rate $\boldsymbol{\theta_3}$ is greater than $\boldsymbol{\theta_1}$ and $\boldsymbol{\theta_2}$. If the difference between the aggregate prediction of the US and the actual US at a given time cannot be reduced by modifying the direct $\boldsymbol{(CS - US)}$ associations, sensory units strengthen the connections to hidden units. That is, the sensory unit gets configured to solve the problem at hand [6] this is a key feature of occasion setting. The SD model is able to handle the regularities of occasion setting, where unique cues seem to modulate responding to common cues rather an acquiring the ability to generate CR by itself [90, 91, 92]. Configural and direct associations with the output unit are regulated by the same error (EO). Successive arrangement of features comes to set the occasion for the response controlled by the target while in a simultaneous arrangement, features directly control the response. The model differential predictions of both simultaneous and serial arrangement of feature positive and feature negative are controlled by the initial values of $C_{i,j}$.

2.0.1.14 Modulatory models

Modulatory models extend the US representation beyond its role as a 'supervisor' to a modulatory function. This means that the US plays a role of modulating the process of acquisition in addition to the role of reinforcement or error computation. The computational modulatory model [93], the Grossberg Model, assumes the CS forms associations with the sensory aspect and with the motivational aspect of the US such that pairing with CS generates changes in two types of associations. First, between the sensory representation of the CS and the drive representation of the US. The CS acquires the ability to elicit the response and becomes a secondary reinforcer. Secondly, associations between the drive representation of the US and a secondary sensory representation of the CS. The US modulates the degree of activation of the CS. The first association is known as conditioned reinforcement, as the CS acquires the ability to act as the US whereas the second association is known as the incentive learning as the activation of the US representation modulates the degree to which the CS will be processed.

VET model [94, 95, 96] was proposed to describe the temporal properties of the conditioned response. This model assumes that during acquisition, two separate but interacting processes take place. First, there is the acquisition of responding in the presence of the CS, and second, the acquisition of appropriate expectancies of the US. Animals learn which response to emit and separately learn when to emit a given response. A dual representation of the US was proposed and implemented [95], with one responsible for CR and the other responsible for a temporal expectancy of the US. Both Grossberg and VET models have little empirical evidence to support the hypothesis.

Wagner and Brandon [97] proposed AESOP, an affective extension of SOP [83, 84]. In this model, the CS forms separate associations with two kinds of US representation, sensory and emotive attributes. The association of the CS with the sensory representation evoke conditioned responses. In contrast, associations with the emotive representation evoke a diffuse of activity changes known as the conditioned emotive response (CER), which modulates the CR. Additionally, some CSs are more likely to be associated with the sensory or emotive aspect of the US depending on the temporal arrangement of the cues in conditioning [6]. Divergence of response measures [98] motivated the need to incorporate an emotive mechanism to SOP. The conditions in which CSs are associated with the US is that CER is more likely to occur with extended (or contextual) CS, whereas CR tends to develop with short duration CS [99, 100, 101, 102]. The nature of the US modulation is such that CER are relatively generalized emotional states that affect performance with any CS. Depending on the nature of CER (appetitive or aversive), they not only modulate the expression of CR by inhibiting or facilitating the response but also modulate the acquisition of CR [103]. Experiments conducted [104, 97] conclude that conditioned response and conditioned emotional response may be controlled by different aspects of the US, and the nature of the CS-US associations depends on the temporal requirements of each type of association.

2.0.1.15 Componential models

We have seen that the CR is affected by several variables brought about by changes in the amplitude and frequency of the response across the course of conditioning. Similarly, temporal factors affect onset latency, peak amplitude and time of peak [73]. The correlation of these factors coupled together can be regarded as a single construct known as the strength of association.

There are instances where these measures become uncorrelated, suggesting a complex mechanism underlying acquisition. For instance, a shift in ISI leads to an observed shift in the latency of CR initiation and peaks to a longer or shorter value [105, 106]. The change in timing is reflected not in a gradual movement of the CR in time towards its new location but in the extinction of the CR at its original locus and reacquisition at its new locus. The strength notion incorrectly predicts the gradual change in CR topography with an ISI shift, reflecting the loss or gain of associative strength expected. Additionally, mixed training at two different ISIs produces two CRs, each resembling the CR exhibited by animals when trained with each ISI separately. The strength notion assumes that the topography of the CR depends entirely on the associative strength accrued to the CS. Mixed training with two ISIs should produce only one CR with a topography that look like the average topography of CRs that normally would develop when each ISI is trained separately.

These challenges have been solved by building more complex, molecular or componential CS representations. This notion assumes that the stimulus trace is made up of a number of components that occur at different moments in time following CS initiation and control separate associations. Consequently, different CR develops with Independence at different ISI, solving the problem of double-peaked CR and the shifting ISI. It is also neuro-biologically plausible to represent the CS as an activity of constellations of neurons that exhibit plasticity at the level of the synapse, evidence of the major temporal properties of Pavlovian conditioning [107, 108, 109].

There exist several ways to construct the componential stimulus. A CS representation such as the tap-delay line suggested by David and Moore [95] where the CS onset initiates a sequence of activations of elements across a temporal line. Grossberg and Schmajuk [93] suggest a CS generated by bell-shaped signals that vary in rise and delay. Gluck [110] suggests a CS representation in which elements behave as sine waves oscillating at different frequencies. Mauk [111, 109] suggested that the CS influences the pattern of activity of neurons whose on-and-off status is determined by a random stochastic process. A dual representation of the CS with two classes of binary CS elements was suggested [112]. In this design, some CS elements have a distribution pattern of activity over the duration of the CS duration, and some have a randomly distributed pattern.

Temporal difference (TD) [113, 114, 79] was an improvement of the SB model. It was proposed to solve the inhibitory effect of the US offset and generation of an ISI function for delay conditioning. This model incorporates a componential representation of the CS. Initially, the TD model

[114, 79] suggests the CS and the US traces were defined exactly the same as the ones proposed in the SB model with both having an eligibility trace and a stimulating trace. Moore et al. [113] proposed a CS representation called the tap-delay-line in which the CS initiates a cascade of sequentially activated elements ordered in line of activation. The CS is assumed to have two traces, an onset and an offset one which contains a sequence of activation elements in each. Each element x_{ijk} has a tap that influences the activity of the adaptive unit. Elements in the delay line are either active or inactive. The activation period is equal for all the elements and lasts several time steps. Once the CS activates the first element in the delay line, the first element activates the second until the last element is activated. The same is applied during the CS offset. Desmond and Moore[95] argued that separated representations of the onset and offset of the CS are necessary because the offset of a stimulus during the inter-trial interval can serve as an effective CS.

Each element is represented by two distinctive traces: the stimulating trace is a binary function, and the eligibility trace reaches a maximum value and determines the rate of learning at each moment in time. Changes in the V_{ijk} is given by:

$$\Delta V_{ijk} = \alpha\beta[Z(t)\lambda(t)+\gamma\sum x_{ijk}(t)V_i(t)-\sum x_{ijk}(t-1)V_{ijk}(t-1)] \tag{2.25}$$

where γ is the discounting factor $(0 < \gamma < 1)$ that allows CS$_i$ with existing associative strength to generate changes in V$_i$. The contribution of the US in this model contributes to reinforcement throughout rather than during the change in intensity (at time $(t-1)$); the reinforcement is determined at the moment (time t) rather than by the change in intensity levels. Additionally, the contribution of the CS to the reinforcement term is weighted by parameter γ that permits the CS to contribute to reinforcement even when the associative strength does not change from time $(t-1)$ to time (t).

Results of implementing the TD model have resulted in alignment with empirical evidence where: an inverted-U shape function relating strength of conditioning and ISI, for both trace and delay conditioning, was plotted; maximal learning at ISIs of intermediate duration; excitatory learning for simultaneous conditioning; and excitatory learning for short ISIs in both forward trace and delay conditioning. Training with large numbers of trials predicts that the ISI function will become linear with greater conditioning for shorter ISI, as with the RW model. This is because the TD model computes inhibitory learning for all parts of the trace in which US is not present, and CS has some excitatory value [6]

Componential SOP

In an SOP model, a stimulus function is described in terms of activity states. Each element is assumed to be in a state of inactivity from which it can be activated, a state of activity from which it can become refractory, and finally, a state of refractory from which it cannot be activated but will revert to inactivity. This becomes a componential representation by assuming that learning accrues with relative independence to the stimulus that is differentially activated during the duration of the CS. The stimuli characterization is assumed to generate the same molar trace from trial to trial based on a probabilistic-determined course of activity. CS representation that is temporally distributed constitutes randomly distributed elements and temporally distributed elements that can account for previously accounted phenomena and CR timing [112].

The idea of componential representations was adopted by [115, 116] in the Spectral Timing Model where the CS activates at different rates, some fast and some slow. The joint activation of all elements belonging to the same population is called the activation spectrum. When joined to the US activation spectrum, there is temporally associative learning [116].

2.0.1.16 Double Error Dynamic Asymptote Model

Double Error Dynamic Asymptote Model[10] (DDA) model is a formal, fully connected computational model of Pavlovian conditioning.

Stimulus representation

In this model, the stimulus representation is a set of elements denoted by e that can be unique to the stimulus or shared with other stimuli. Shared elements are sampled whenever one of their parent stimuli is active. Additionally, this model constitutes an elemental framework of stimulus representation where each element of a given temporal cluster of a given stimulus is capable of developing associative links to other temporal structures.

These elements learn to predict clusters of other elements. This model implements differential activity through time, where elements in a cluster vary in activity through time. Any stimulus can either function as a predictor, $\boldsymbol{p}$, or an outcome, $\boldsymbol{o}$, that contain a set of clusters $\boldsymbol{i, j}$

For each time unit of the duration of the stimulus, a temporal cluster is defined with the maximal cluster activity happening at time $\boldsymbol{\bar{t}}$. Each element within each cluster is activated with a probability that approximately follows a Gaussian distribution at that moment in time. This is depicted as:

$$\begin{aligned}
&\forall e \ \vee \ (i, \ p \vee o) \\
&\Phi^{t}_{i,p} = exp(-\frac{(t-\bar{t})^2 k}{2\sigma_i^2}) \cdot (I \; iff \; p \vee o = US)
\end{aligned} \tag{2.26}$$

Where $\boldsymbol{\sigma_i^2 = CVg\bar{t}_i}$ is the variance of the temporal cluster. CV depicts the coefficient of variation and g. The model assumes that $\boldsymbol{\bar{t}_i = i}$, which implies that the $\boldsymbol{i}$'th temporal cluster peaks at time point $\boldsymbol{t = i}$. $\boldsymbol{k}$ is a skew parameter that multiplies the enclosed term when $\boldsymbol{t < i}$. If the cluster $\boldsymbol{i}$ belongs to a US, the scalar intensity value $\boldsymbol{I}$ multiplies the equation such that the strong reinforces are coded with high $\boldsymbol{I}$ values. The default value of I is set to 1.

The DDA model uses temporal clusters to implement the variable stimulus representation in time and differential element activity during the presentation of the stimulus. Earlier temporal clusters learn associations differentially from later ones. This mechanism ensures that there is significant generalization through time, as thereis usually an overlap of activations between long tails of the temporal clusters. The DDA model implements a different eligibility mechanism to counteract inter-trial extinction.

Associative Activation

The DDA model evokes activity without a direct source of activation. In general, The activation of an element results in the prediction of clusters of elements proportional to the weights from itself to these clusters. At a given time-point, the aggregate of the predictions for a cluster i functioning as an output o of another cluster j, denoted by $\Psi^t_{i,o}$ is the total prediction for this cluster i, consisting of the contribution of each predicting element,e, active at that point in time $(x^t_{e,i\vee j,p\vee o} = 1)$. That is, the associative activation of a cluster is a function of the weight of the link from each active predicting element at that moment in time $(w^t_{e,j,p\rightarrow i,o})$ modulated by $0 < \vartheta < 1$, an associatively activated discount that is triggered if the predictor j is not directly activated but retrieved. The associative activation of a cluster is computed as:

$$\Psi^t_{i,o} = \sum_p \sum_j \sum_e w^t_{e,j,p\rightarrow i,o} x^{t-1}_{e,i\vee j,p\vee o} \cdot (\vartheta \; if \; \Phi^t_{j,p} < 0.1 \; else \; 1) \tag{2.27}$$

where probability $(x^t_{e,i\vee j,p\vee o} = 1) = A^t_{i,p\vee o}$. Where $A^t_{i,p\vee o}$ is the overall activation. The notion $\Psi^t_{i,o}$ is used to indicate that associatively activated elements are predicted (i.e., the outcomes o)

Overall Activation

The overall activation of the cluster, that is either a predictor or an outcome $(\boldsymbol{i, p \vee o})$ is taken to be whichever is larger between its associative activation, $\boldsymbol{\Psi^t_{i,o}}$ or direct activation $\boldsymbol{\Phi^t_{i,p \vee o}}$. This is expressed as:

$$\boldsymbol{A^t_{i,p \vee o} = max(\Phi^t_{i,p \vee o}, \Psi^t_{i,o})} \tag{2.28}$$

The activation of a cluster, direct or associative, complements rather than competes against each other. This implies that predictions by one stimulus to another do not inhibit the activation of the predicted stimulus rather, they reduce the novelty of the predicted stimulus.

Learning and the Dynamic Asymptote

Learning is set to occur when two clusters are concurrently active, either through sensory experience or associative retrieval. The direction of learning (excitatory or inhibitory) between clusters is dependent on the dynamic asymptote, which is a measure of closeness between the aggregate activation of the clusters, as well as the associative strength of other cues in the error term. A unique feature of the dynamic asymptote is that it predicts within-stimulus learning while being able to produce stimulus-to-stimulus learning. The excitatory and inhibitory links are derived from the error term of the outcome.

The asymptote of learning used in the outcome error term is an inverse measure of the distance in activity between the predictor cluster and the predicted cluster. The asymptote for each cluster is estimated using a constrained overall stimulus activation $\boldsymbol{\hat{A}}$, such that if the stimulus direct activation is above a threshold of 0.1, the value is set to maximum direct activation.

$$\hat{A}^t_{i,p\vee o} = \begin{cases} max\Phi^t_{i,p\vee o}, & \text{if } \Phi^t_{i,p\vee o} > 0.1 \\ A^t_{i,p\vee o}, & \text{otherwise} \end{cases} \tag{2.29}$$

$$\text{where}(max\Phi^t_{i,p\vee o} = I\text{if}(i \in US)\text{else}1)$$

$$\lambda^t_{i,p\rightarrow j,o} \triangleq \| \hat{A}^t_{i,p}, \hat{A}^t_{j,o} \| = \frac{\hat{A}^t_{j,o} - |\hat{A}^t_{j,o} - \hat{A}^t_{i,p}|}{max(\hat{A}^t_{j,o}, \hat{A}^t_{i,p})} \tag{2.30}$$

The result is an asymptote based on a linear distance function, where two cues with highly dissimilar activations support less learning than if their activations at a given time point were similar. As the absolute value is subtracted from the outcome total activity level, this dynamic asymptote is anti-symmetrical; that is, the outcome activity is more determinant of whether the asymptote is positive or negative.

The Double Error Term and Weight Update

The outcome error is given by the difference in the asymptote of learning $\lambda^t_{p\rightarrow j,o}$ and the total prediction of the outcome $\Psi^t_{j,o}$ depicted by:

$$\forall\, e \in (i, p)$$

$$\delta^t_{i,p\rightarrow j,o} = \lambda^t_{i,p\rightarrow j,o} - \Psi^t_{j,o} \tag{2.31}$$

In the DDA model, the weights encode the degree to which the activity of the predicting element predicts the activity of the outcome cluster. One of the novel features of this model is that the predicting cluster has an error term that denotes how expected the predictor stimulus is. This error term is used to modulate learning in the model as well as to define the reevaluation of the alpha update.

$$\forall\, e \in (i, o)$$

$$\delta^{t}_{\rightarrow i,o} = \| \; A^{t-1}_{i,o} - \Psi^{t}_{i,o} \tag{2.32}$$

Both the predictor and the outcome error are used to update the weights for a given element of a given temporal cluster.

$$\Delta w^{t}_{e,i,p\rightarrow j,o} = \delta^{t}_{i,p\rightarrow j,o}\delta^{t}_{\rightarrow i,o} s_{e,i} s_{j} x^{t}_{e,i\vee j,p\vee o} A^{t}_{i,p} A^{t}_{j,o} \epsilon^{t}_{i,p} \alpha^{t}_{i,p\rightarrow c} \cdot (b \text{ iff } \bar{t}_i > \bar{t}_j) \tag{2.33}$$

Where $s_{e,i}$ and s_j are saliences of the predicting element and predicted cluster, respectively. $A^{t}_{i,p}$ and $A^{t}_{j,o}$ are activations while $x^{t}_{e,i\vee j,p\vee o}$ is a binary activity term and $\epsilon^{t}_{i,p}$ is the eligibility term used to counteract extinction before outcome occurrence. $\alpha^{t}_{i,p\rightarrow c}$ is the adaptive re-evaluation rate. b is a backward discount factor that multiplies learning from $i \rightarrow j$ if $\bar{t}_i > \bar{t}_j$, i.e if cluster i occurs after cluster j.

The direction of learning is determined by the outcome prediction error and its variable asymptote, while the prediction error for the predictor itself, along with other modulating factors, influences the extent and speed of learning.

Eligibility Modulation

The DDA model is implemented as a real-time model, and outcomes are predicted before their onset hence significant extinction occurs on absent outcomes. The eligibility factor counteracts this trend. The observed prediction from one cluster to another is given by

$$\boldsymbol{\Psi}^{t}_{i,p \rightarrow j,o} = \sum_{e} w^{t}_{e,i,p \rightarrow j,o} x^{t-1}_{e,i \vee j,p \vee o} . (\vartheta \text{ if } \boldsymbol{\Phi}^{t}_{i,p} < 0.1 \text{ else } 1) \tag{2.34}$$

The eligibility is defined as operating on cluster-to-cluster temporal predictions. When the predictor cluster is not present but associatively retrieved, then the observed prediction $\boldsymbol{\Psi}^{t}_{i,p \rightarrow j,o}$ is multiplied by ϑ. The eligibility is expressed as:

$$\epsilon^{t}_{i,p} = (\frac{\boldsymbol{\Psi}^{t}_{i,p \rightarrow j,o}}{max \boldsymbol{\Psi}^{t}_{i,p \rightarrow j,o}})^{z} \tag{2.35}$$

For an active predictor, p, the $max \boldsymbol{\Psi}^{t}_{i,p \rightarrow j,o}$ for each outcome is updated towards a maximal prediction for that outcome cluster in the current trial (T). The rate of the update is determined by the eligibility discount γ expressed as:

$$max\Psi^{t}_{i,p\longrightarrow j,o} = max\Psi^{t-1}_{i,p\longrightarrow j,o}\gamma + (1-\gamma)max\Psi^{T}_{i,p\longrightarrow j,o} \tag{2.36}$$

Attentional modulation: The stimulus Associability

Attention to cues increases when there is uncertainty in the occurrence of an outcome. If the uncertainty occurs for a long period of time, this is used as a source of information and attention levels to a cue are reduced. The attentional modulation is proportional to the time-dependent activation $^{t}_{i,p}$ of the element as well as a fixed adaptation parameter, ρ, that determines how quickly the re-evaluation changes. This is depicted by:

$$\alpha^{t}_{i,p\longrightarrow c} = (1 - d^{t}_{i,p\longrightarrow c})(1 - \rho A^{t}_{i,p})\alpha^{t-1}_{i,p\longrightarrow c} + \rho A^{t}_{i,p}|\delta'^{t}_{i,p\longrightarrow c}| \tag{2.37}$$

where $\delta'^{t}_{i,p\longrightarrow c}$ is the overall error of the class of outputs calculated as a moving average in time and $d^{t}_{i,p\longrightarrow c}$ is a decay that is initiated if the moving average crosses a given threshold.

2.0.2 Convolution Neural Networks - CNN's

2.0.2.1 Introduction and Background

The ability of humans and animals to process image objects locatable in time and space, with many features, is supported by significant specialized neural mechanisms. The brain processes a visual world through the intake of sensory outputs from the neural transducers. This stimulation results in the activation of a complex network of neurons [117] and hence the formation of mental representations. These representations are built up sequentially from a hierarchy of interconnected areas in the visual cortex, as low-level area representations contribute to the building blocks of high-level representations.

Convolution Neural Networks (CNNs) [118], a class of multilayer computational models collectively known as *convolution networks* (ConvNets), specialize in processing grid-like data such as time-series data and image data [119]. These models have experienced tremendous success in practical tasks such as object recognition, segmentation, regression classification etc. [120, 121, 118]. CNNs make use of a mathematical operation known as convolution, a specialized kind of linear operation that leverages sparse interactions, parameter sharing, and equivariant representations to produce a set of linear activations [119]. Each linear activation function is run through a non-linear activation function, such as a rectified linear unit - (ReLU) that detects features of interest. Finally, a pooling operation is applied, which modifies the output further, making representations invariant to small translations.

End-to-end CNNs are trained using supervised learning methods to associate visual input representations with appropriate labels [122]. After training, they generalize to unseen images from a given test set [123]. Although ConvNets have existed for a long time, several state-of-the-art machines for image recognition in vision and learning have been developed mainly influenced by the ongoing deep learning revolution [121].

The ImageNet challenge [123], since 2010, has seen a rise in the development and improvement of traditional computer vision models for image recognition to the deployment of state-of-the-art models such as ResNet deep learning framework for training Deep CNNs. These architectures have managed to achieve human-level performance in object recognition [120, 124, 125, 126].

CNNs are characterized by deep convolution layers containing thousands of connection weights. Their structural design mimics the hierarchical structure of the visual cortex, containing sequential information encoded in lower areas flowing to higher levels with representations characterizing the information encoded about the visual stimuli [5]. Activations induced by stimulus spread sequentially up the hierarchy, where each layer computes new representations as encoded in the feature maps. The mechanisms of feature detection along the visual cortex follow a neurobiological motivation that stems from the pioneering works of David Hubel and Thorsten Wiesel [127] [128] who focused on locally sensitive and orientation-selective neurons of the visual cortex of a cat. Results of the recording from V1 cells found two types of cells: **Simple cells** that responded to bar-like patterns at a particular orientation and position. **Complex cells** responded to the bars and had a preferred orientation but had a small degree of position invariance [5].

Our interest is to use CNNs as computational models, which will enable us to explore how representations are identified along the ventral streams. We are interested in the use of these models to harness the power of encoded information to account for association formation as representations are built up along the hierarchical areas.

2.0.2.2 Structural description of a Standard CNN

A CNN model is composed of successive layers that mimic the hierarchical organization of the visual cortex. The network employs a linear mathematical operation known as convolution, followed by applying a non-linear activation function and, lastly, applying a pooling operation on data. These operations are typical and essential in a standard CNN layer as they enable information about features to be encoded to characterize representations in the layer.

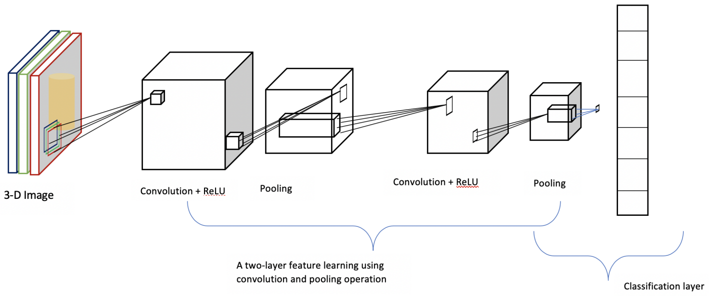


Figure 2.1: A classical architecture of a Convolution Neural Network (CNN). Each layer is composed of convolution and pooling operations.

2.0.2.3 Convolution Operation

Convolution is a mathematical operation applied to two real-valued arguments, say x and t. Suppose we would like to track the location of an object x at a given time interval t using a laser. To get the reading of the position at a given time t, we average several measurements such that the weighted average value gives more weight to the more recent measurement. The weighting function is defined by $w(a)$. Application of the average weighted operation at every moment in time results in a new function given by:

$$s(t) = \int x(a)w(t-a)da \tag{2.38}$$

This operation is known as convolution, and its mathematical denotation is given as:

$$s(t) = (x * w)(t) \tag{2.39}$$

In a CNN network, the function x is referred to as the input, and the second argument is the kernel, which is sometimes often referred to as a feature map. These input and kernel functions are multidimensional arrays of data parameters to be learned [119].

Given a two-dimensional image I, a two-dimensional kernel K is implemented such that the new function (feature map) is given by:

$$s(i,j) = (I * K)(i,j) = \sum_{m} \sum_{n} I(m,n)K(i-m,j-n) \tag{2.40}$$

One of the properties of convolution is that it is commutative; therefore, it can be written as:

$$s(i,j) = (I * K)(i,j) = \sum_{m} \sum_{n} I(i-m,j-n)K(m,n) \tag{2.41}$$

This implies that we are convolving the image function I, containing pixels locations i,j with a kernel function over a range of valid values m and n. The cross-correlation function is the same as convolution but without flipping the kernel [119]. Whereas both operations achieve similar results, many machine learning libraries implement convolution as cross-correlation [119].

$$s(i,j) = (I * K)(i,j) = \sum_{m} \sum_{n} I(i+m, j+n)K(m,n) \quad (2.42)$$

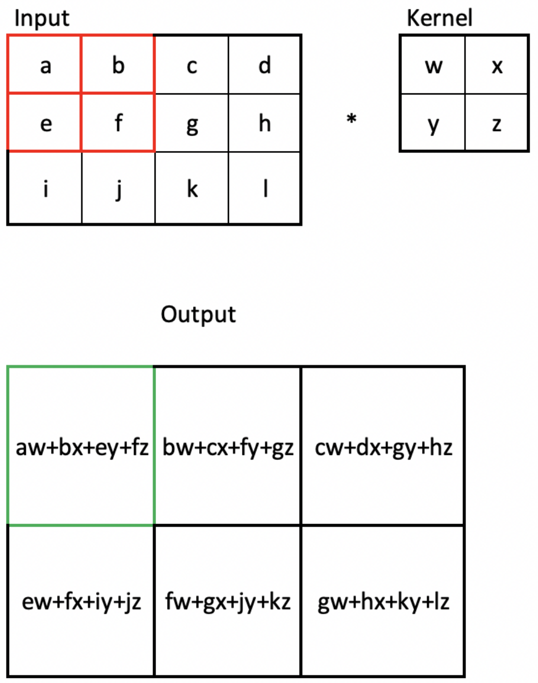

Input
a b c d
e f g h
i j k l
*
Kernel
w x
y z
Output
aw+bx+ey+fz bw+cx+fy+gz cw+dx+gy+hz
ew+fx+iy+jz fw+gx+jy+kz gw+hx+ky+lz

Figure 2.2: A 2D convolution without flipping the kernel applied in a 2D Image.

In a CNN, each neuron from the feature map is only connected to a local region of the input. It is impractical to connect all neurons as images are high-dimensional inputs. Suppose there is a (10x10x3) image fully connected to a single unit with 300 weights. Convolution enables local connections of learnable filters that are moved across the height and width of the image to produce an activation map. Simple arrays of neurons in the feature maps with similar properties (the same weights) are arranged spatially such that copies of the same kernel are found in different positions in space. The filter is smaller than the inputs, and therefore, sparse interactions are established [119]. The image I is convolved by a filter/kernel w. The output at each pixel is given by the product of the filter to the appropriate intensity values of the image that produces the output of the filters at every spatial position and an additional bias term.

$$\boldsymbol{y = \sum_{i \in 3X3} w_i x_i + b} \tag{2.43}$$

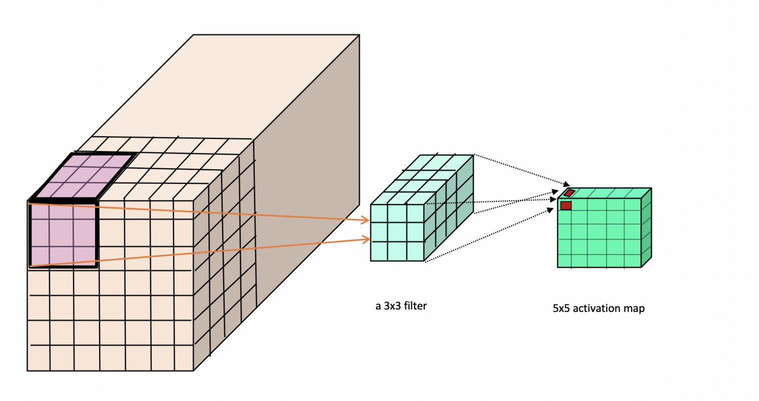


Figure 2.3: Simple diagram depicting locally connected high dimensional image (7x7x3) to a 3x3 filter and a5x5 activation map

The output is controlled by three hyper-parameters, depth, stride and padding. Depth corresponds to the total number of filters (in this example, there are three filters convolving each dimension of the image). Each filter corresponds to the 3 colour channels of the image. Each neuron in the activation map is connected to the receptive field (a 3X3 filter in this case). The connections of the filter map are connected by a matrix of weights (kernels) that are shared. This reduces the number of parameters of the model, and the resulting output becomes equivariant. A feedforward pass of the image activates the neurons along the depth dimension in the presence of various oriented shapes, colours or other specified features.

Convolving through the image involves sliding the kernel across the image. This operation may involve techniques such as selecting a given stride which allows the flexibility to specify how to slide the filter or the kernels across the image. Convolving also involves applying padding the input with zeros around the border. Padding enables the preservation of the spatial size of the output such that input and output sizes are the same or sometimes smaller or larger than the original input.

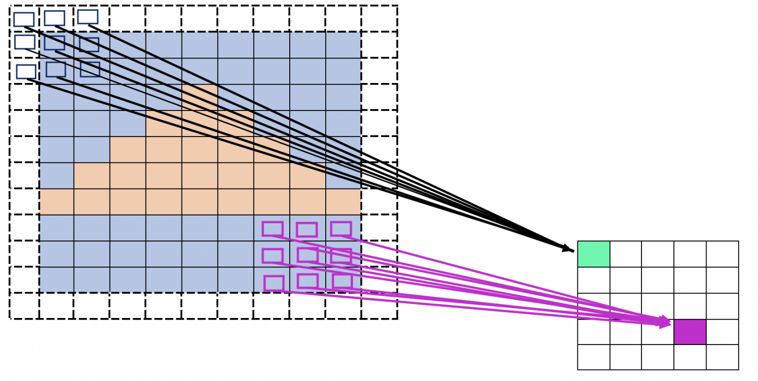


Figure 2.4: Simple diagram depicting locally connected high dimensional image (7x7x3) to a 3x3 filter and a5x5 activation map

2.0.2.4 Variants of the Convolution Operation

Valid Convolution

This is the simplest convolution operation. It entails fully overlapping the kernel with the image such that the output will be slightly smaller than the original input. The output is given by:

$$\boldsymbol{outputsize = inputsize - kernelsize + 1} \tag{2.44}$$

Full Convolution

This is the opposite of a valid convolution. Wherever the kernels and the image overlap by at least one pixel, an output is computed. In practice, this operation ends up padding the image with either zeros or any other values at the discretion of the modeller. The activation feature map computed ends up being larger than the original image. Full convolution can be thought of as a valid convolution but only with padding. The output size is given by:

$$\boldsymbol{output_size = inputsize + kernelsize + 1} \tag{2.45}$$

Same Convolution

In this operation, the image is padded with enough zeros so that the feature map is the same size as the original input. The padding also depends on the kernel size. For instance, if the kernel size has an even number dimension, the input image will be padded asymmetrically. One thing to look out for is the possibility of detecting edges of the input as important features extracted by the image. The output size is given by:

$$\boldsymbol{outputsize = inputsize} \tag{2.46}$$

Strided Convolution

In this operation, the output is computed by skipping some steps instead of computing the output at every possible offset, which results in the feature map becoming smaller than the original input. The **??**below shows a strided convolution with a two step sliding kernel. This operation reduces the resolution of the feature maps and consequently, this enhances high-level features in the hierarchy to be operating at a larger scale. Additionally, this technique is a cheaper way to compute.

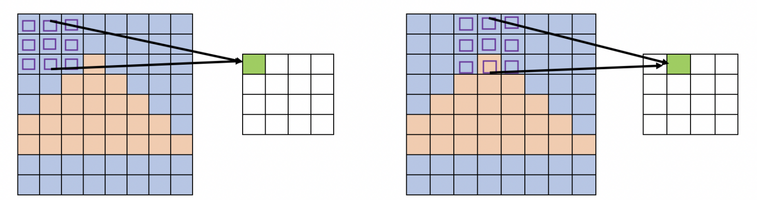


Figure 2.5: Simple diagram depicting a two step Strided convolution operation

Dilated Convolution

The operation skips the value of the receptive field. It results in varying the features of the images slowly over space by sub-sampling. While increasing the size of the filter is an alternative, it remains an expensive decision as this results in increased parameters and heightens the computation cost.

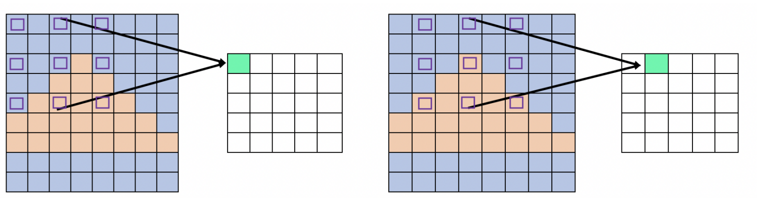


Figure 2.6: Simple diagram depicting a dilated convolution operation

2.0.2.5 Pooling Operation

The pooling function replaces the output of the unit at a certain location with a summary statistic of nearby neighbours [119]. Pooling layers functionally reduce the spatial size of the feature maps. Representations invariant to small translations of the input are extracted. Additionally, the Pooling function operates independently on the feature sheet of the input and resizes it spatially, using the MAX operation [129]. It sums up similar information in the neighbourhood of the receptive field and outputs the dominant response within this local region.

$$\boldsymbol{Z_l^k} = \boldsymbol{g_p}(\boldsymbol{F_l^k}) \tag{2.47}$$

Pooling operation is defined as $(\boldsymbol{Z_l^k})$, which represents the pooled feature-map of the $\boldsymbol{l}^{th}$ layer for the $\boldsymbol{k}^{th}$ feature map, whereas $\boldsymbol{g_p}(.)$ defines the type of pooling operation. The use of pooling operation helps to extract a combination of features, which are invariant to translational shifts and small distortions [130, 131] Reduction in the size of feature-map to invariant feature set not only regulates the complexity of the network but also helps in increasing the generalization by reducing overfitting [132]. In addition to max pooling, the pooling units can also perform other functions, such as average pooling, which is viewed as adding an infinitely strong prior such that the function the layer learns is invariant to small translations [119].

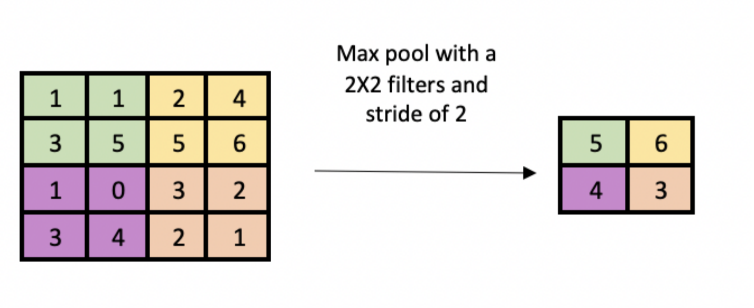


Figure 2.7: Simple diagram depicting a max pool operation

2.0.2.6 Activation Function

The activation function serves as a decision function and helps learning about intricate patterns. This function is applied after convolution for a convolved feature map and it is defined as:

$$\boldsymbol{T_l^k} = \boldsymbol{g_a}(\boldsymbol{F_l^k}) \tag{2.48}$$

The output of a convolution $(\boldsymbol{F_l^k})$ is passed through an activation function $\boldsymbol{g_a}(.)$ that adds non-linearity and returns a transformed output $\boldsymbol{T_l^k}$ for the $\boldsymbol{l^{th}}$ layer [132]. Different activation functions such as sigmoid, tanh, ReLU, and variants of ReLU, are used to apply a non-linear combination of features [133, 134, 135, 136] However, ReLU and its variants are preferred as they help in overcoming the vanishing gradient problem [137, 132]. One of the recent activation functions MISH: Self Regularized Non-Monotonic Neural Activation Function, has been shown to perform better than ReLU[138]

The success of CNNs, therefore can attributed to these structural designs that apply three forms of constraints [139]

- Feature Extraction

  Units are organised into planes, also known as *Feature Maps*. Each unit takes inputs from a small sub-region of the image - *local receptive field*. These units detect the same patterns but at different locations in the input image. Once the features have been extracted, location becomes less important but more importantly, there is preservation of position relative to other features.

- Feature Mapping

  There are multiple feature maps, each having its own set of weights and bias parameters. They are all constrained to share the same synaptic weights. As a result, shift invariance is incorporated into the operation of the network through the use of convolution followed by a sigmoidal non-linearity activation function.

- Subsampling

  The outputs of the Convolution units form the inputs of the pooling layer. A plane of units is formed in this layer. Each unit takes inputs from a receptive field that performs an operation such as local averaging of those inputs. These are multiplied by adaptive weights and the addition of an adaptive bias parameter and then transformed by the application of a sigmoidal non-linearity activation function. This operation reduces the sensitivity of the feature map outputs to shifts and other forms of distortion.

2.0.2.7 CNN's Variants

CNNs date back to the late 80's when LeCuN [118] proposed a 2D supervised CNN model that performed tasks such as reading handwritten digits and zip codes [140]. Different improvements of the CNN architecture have been deployed to date that can perform tasks such as image classification and segmentation, object detection, video processing, natural language processing, and speech recognition. The improved capability and performance in these models can be attributed to parameter optimization, structural reformulation, application of different regularization techniques etc [132]. Most of the improvements made are due to a restructuring of the processing units and the innovation of new blocks(layers). The attributes of some of these modifications are categorized into seven categories [132], namely, spatial exploitation, depth, multi-path, width, feature-map exploitation, channel boosting, and attention-based CNNs. We will review these architectures briefly while stating examples of the CNNs architectures that have been developed.

2.0.2.8 Spatial Exploitation based CNNs

Parameters and hyper-parameters in a CNN such as weights, biases, activation function, learning rate, filter size, strides, padding, number of processing units, and number of layers characterize the CNN model [141, 142]. These models focus on exploiting the locality of connectivity of the pixels in an image. Different filter sizes have been explored to improve the performance and the learning of the network. Research has suggested that adjustment of filters can improve the granularity of detail and information detected by the filterers [132]. Some of the architectures under this category are *LeNet* [139, 143], which displayed state-of-the-art performance on hand-digit recognition tasks. The architecture classified digits without being affected by small distortions, rotation, and variation of position and scale.

AlexNet [120] displayed groundbreaking results for image classification and recognition tasks. This model improved the learning capacity of the CNN by making it deeper and by applying several parameter optimisation strategies [120]. The hardware limitations experienced in the early 2000s were never experienced with this architecture. It was trained using two parallel GPUs (NVIDIA GTX 580) to overcome its shortcomings.

ZfNet, also known as a multilayer Deconvolutional Neural Network (DeconvNet). It was proposed in 2013 [144] to visualize the network performance quantitatively with the aim of monitoring the performance of the CNN by interpreting the neuron's activation. The DeconvNet works in the same manner as the forward pass CNN but they reverse the order of convolution and pooling operation. This reverse mapping projects the output of the convolution layer back to visual image patterns, consequently providing the neuron-level interpretation of the internal feature representation learned at each layer [145, 146]. Feature visualization was also used for the identification of design shortcomings and for the timely adjustment of parameters.

VGG was modular in layer patterns [124] and designed with deeper layers than AlexNet and ZfNet. It was used to simulate the relation of depth with the representational capacity of the network [144, 120] VGG used small filters, and it was discovered that the use of small filters added a benefit of low computational complexity by reducing the number of parameters. These findings set a new trend in research to work with smaller size filters in CNN [132]

GoogleNet [125] achieved a high accuracy with a reduced computational cost. An inception block was introduced whereby multi-scale convolution transformations were incorporated and Split convolutions applied, followed by passing a non-linearity function and then application of the pooling operation. After which, merging was applied. This block convolves filters of different sizes to capture spatial information at different scales. The exploitation of the idea of splitting, transforming, and merging by GoogleNet, helped in addressing a problem related to learning diverse types of variations present in the same category of images having different resolutions. One of the downsides of this architecture was a representation bottleneck that drastically reduced the feature space in the next layer, and thus it led to loss of useful information.

2.0.2.9 Depth based CNNs

These architectures were developed with the assumption that an increase in the depth of the network can better approximate the target function with a number of nonlinear mappings and more enriched feature hierarchies [147]. Although the increase in the depth of the network results in efficient function representations, it comes at the cost of exponentially many neurons. However, it was suggested that deeper networks can maintain the expressive power of the network at a reduced cost [148, 132, 149, 150, 151] Architectural networks such as Inception and VGG, which recorded the best performance, strengthened the idea that the depth is an essential dimension in regulating learning capacity of the networks [124, 125, 152, 153] ResNet [126] also revolutionized the CNN architectural innovation by introducing residual learning and devised an efficient methodology for the training of deep networks. Inception-V3, V4 and Inception-ResNet are improvements of Inception-V1 and V2. Inception-V3 was developed with the idea of reducing the computational cost without affecting generalization. Large size filters (5x5 and 7x7) were replaced with small and asymmetric filters(1x7 and 1x5) and used 1x1 convolution as a bottleneck before the large filters. For Inception-ResNet architecture, residual learning and inception block were combined [126, 152, 153]. Another implementation was Inception-V4 with residual connections (Inception-ResNet). This architecture had the same generalization power as plain InceptionV4 but with increased depth and width. It was observed that Inception-ResNet converges more quickly than Inception-V4, which depicts that training with residual connections accelerates the training of Inception networks significantly.

2.0.2.10 Multi-Path based CNN

CNNs suffer from vanishing gradients or explosions and performance degradation. These challenges are not caused by overfitting. Instead, they are caused by an increase in the depth of the network [137, 154]. Although increasing the depth of the network results in an improvement in performance, the downside is faced while training the network. In deep networks, the large number of layers may result in the backpropagation error computing small gradient values at lower layers. Multi-path networks were proposed to train Deep networks, [155, 156] using multiple paths or shortcut connections that systematically connect one layer to another by skipping some intermediate layers to allow the specialized flow of information across the layers [157]. Cross-layer connectivity partitions the network into several blocks. These paths are an alternative technique for solving the vanishing gradient problem by making gradients accessible to lower layers. These architectures make use of different types of connections, such as zero-padded, projection-based, dropout, skip connections, and 1x1 connections. Architectures that have been designed are highway networks, ResNets[126] and DenseNets [155, 126]

2.0.2.11 Width based Multi-Connection CNNs

As much as the depth of the network is important for efficient representations of complex problems, the width of the network is equally important [158]. Drawn from the use of parallel multiple processing units of the multilayer perceptron, it has been shown that the width is an essential parameter in defining principles of learning [132]. Wide ResNet [159] is implemented to fix a major drawback of feature reuse in deep residual networks where some feature transformations or blocks may contribute very little to learning. Wide ResNet increases the width by introducing an additional factor which controls the width of the network. Widening of the layers provides a more effective way of performance improvement rather than making the residual networks deep. Architectures such as PyramidNet [160] increase the width gradually per residual unit. This strategy enables pyramidal Net to cover all possible spatial locations instead of maintaining the same spatial dimension within each residual block until down-sampling occurs. Another architecture Xception is another example of a width based architecture. It is considered as an extreme Inception architecture, which exploits the idea of depth wise separable convolution [161]. The original inception block are modified in by widening and replacing the different spatial dimensions (for example a 1x1, 5x5, 3x3) is replaced by with a single dimension (3x3) followed by a 1x1 convolution to regulate computational complexity.

2.0.2.12 Feature-Map (ChannelFMap) Exploitation based CNNs

These CNNs, feature selection plays a vital role in determining the performance of classification, segmentation, and detection tasks. Some of the feature-maps impart little or no role towards object discrimination [162]. Enormous feature sets may create an effect of noise and thus lead to overfitting of the network. These networks emphasise on selection of feature-maps to improve the generalization of the network. Architectures such as Squeeze and Excitation Network [162] exist where a new layer called SE-block suppresses the less important feature-maps but assigns a high weight to the class specifying feature-maps. Competitive Squeeze and Excitation Networks is a different form of architecture using the idea of SE block to improve the learning of deep residual networks.

2.0.2.13 Channel-Input Exploitation based CNNs

Filters are applied to extract different levels of information for a single type of image [163, 164]. Since the model relies on input representation, these CNNs rely on the concept of channel boosting (input channel dimension) to boost the representation of the network [165]. The architecture boosts the number of input channels in order to improve the representational capacity of the network. Channel boosting is performed by artificially creating extra channels (known as auxiliary channels) through auxiliary deep generative models and then exploiting it through the deep discriminative models [132]

2.0.2.14 Attention based CNNs

CNNs extract representations that are encoded with different knowledge at different levels of abstraction. In addition to learning about multiple hierarchies of abstractions, focusing on features relevant to the context also plays a significant role in image localization and recognition [132] A practical application of attention based CNNs is the recognition of objects from cluttered backgrounds and complex scenes. Architectures such as residual attention neural network [166] have improved feature representation of the network for learning features. The architecture adopts a bottom-up, top-down learning strategy. The bottom-up feed-forward structure extracts low-resolution feature-maps with strong semantic information. Whereas top-down architecture extracts dense features to make an inference of each pixel [132]. An additional architecture is the Convolutional Block Attention Module [166, 162]. This architecture is simple in design and similar to Squeeze and Excitation-Network, in which the spatial location of the object has a vital role in object detection. Concurrent Spatial and Channel Excitation Mechanism [167] incorporates spatial information in combination with feature-map (channel) information in segmentation tasks.

Chapter 3

Methods

We built the Self Organising Associative Recurrent Network Figure 3.2 that simulates the formation of complex representation structures using associative learning mechanisms. The design pattern that was settled on is summarized in the UML (Unified Modeling Language) diagram below Figure 3.1. This diagram details the overall structure of the modules implemented, which are described in more detail in the Results section since building this very complex model is the main contribution of the thesis.

Difficult trade-off decisions were made to realistically implement the model using standard programming best practices that ensured standard boilerplate code was used to avoid code replication and allow easy tracking of changes.

We used Python as the standard development language and settled on Visual Studio Code as our integrated development environment (IDE). This choice was based on the ease of the language and tools that I was comfortable and familiar with, as well as consideration of the wider community of developers who might want to familiarize themselves with this model. The code for this implementation can be found here. The files inside this package hold 17 modules with 5016 lines of code.

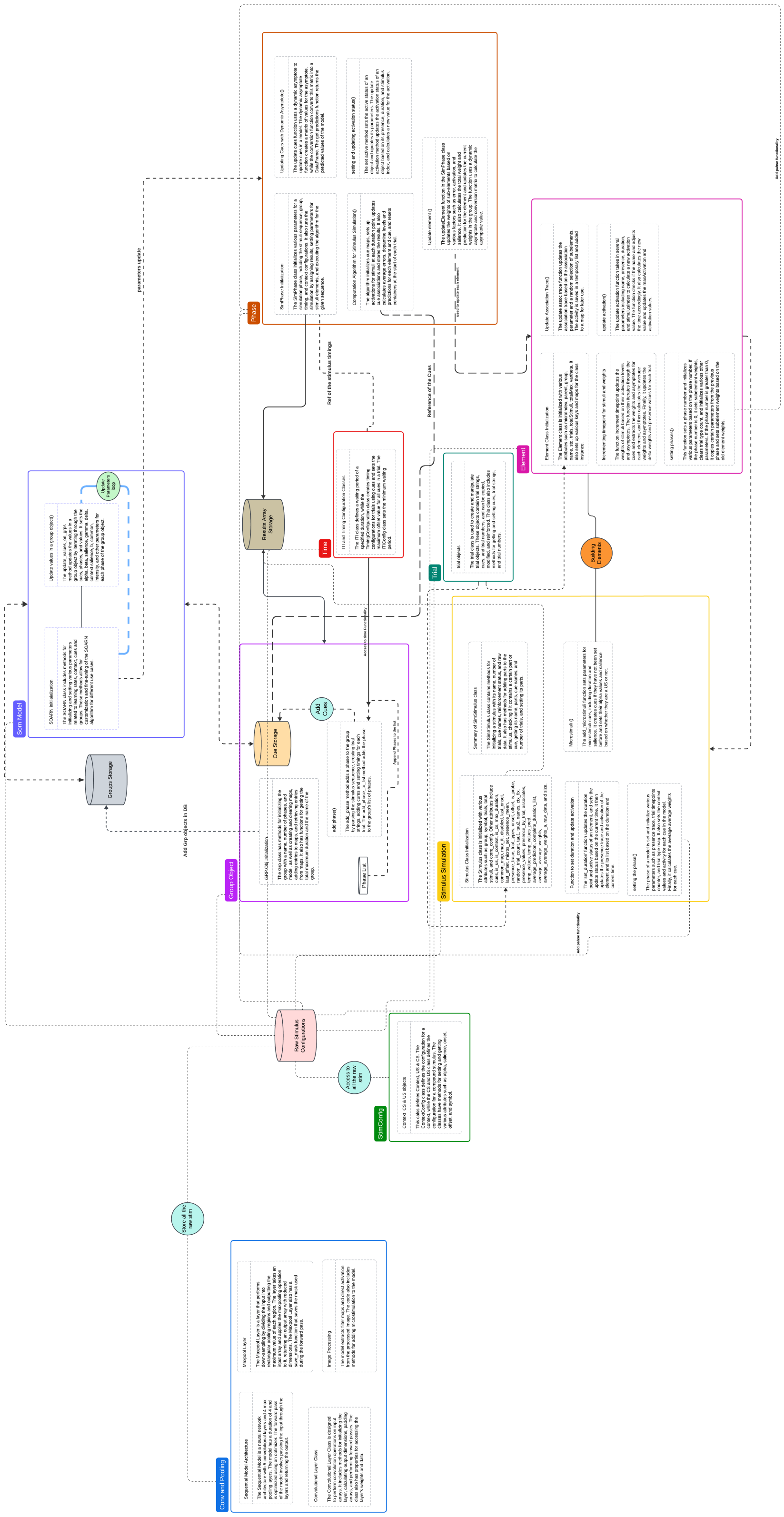


Figure 3.1: Full Architecture UML

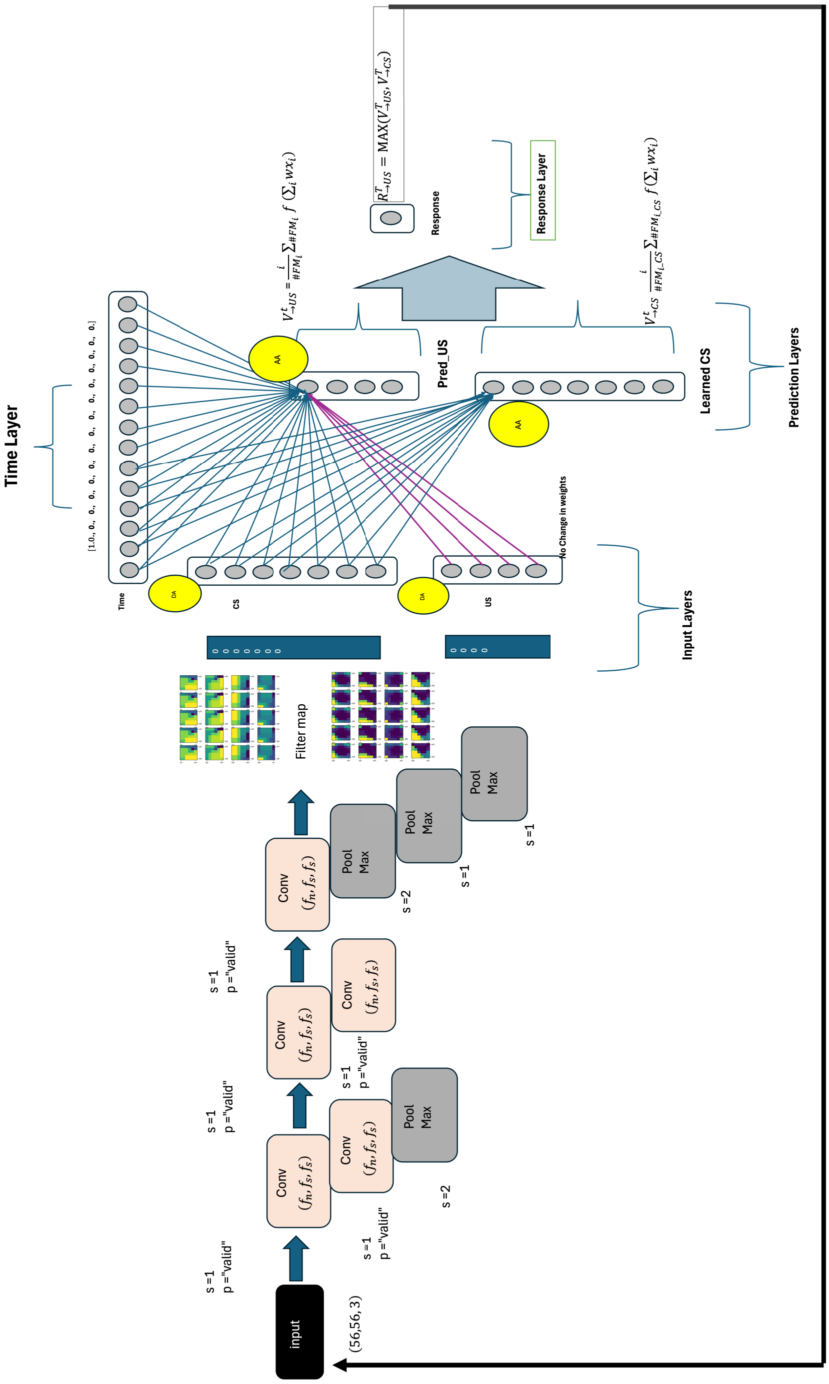

Time Layer
[1.0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]
Time
CS
US
DA
DA
AA
AA
No Change in weights
Input Layers
Pred_US
Learned CS
Prediction Layers
$V^t_{\rightarrow US} = \frac{i}{\#FM_i} \sum_{\#FM_i} f\left(\sum_i wx_i\right)$
$V^t_{\rightarrow CS} \frac{i}{\#FM_{i\_CS}} \sum_{\#FM_{i\_CS}} f\left(\sum_i wx_i\right)$
Response
$R^T_{\rightarrow US} = \text{MAX}(V^T_{\rightarrow US}, V^T_{\rightarrow CS})$
Response Layer
Filter map
input
(56,56,3)
s =1
p ="valid"
Conv
(f_n, f_s, f_s)
Pool
Max
s =2
s =1

Figure 3.2: Full Architecture - SOARN

The file registry contains modules offering different functionalities. We provide a summary of what each module is capable of, noting that there is much more functionality that is abstracted by the descriptions we have provided, and in-depth operational details are described in the next chapter.

3.0.1 Conv

This module has been designed to perform convolution operations on input arrays. It includes methods for initializing the layer, calculating output dimensions, padding arrays, and performing forward passes. The class also has properties for accessing the layer's weights and data.

3.0.2 Phase

This module has been designed to initialize different parameters for a simulation phase, including the stimuli sequence, group, timing, and context configurations. It also runs the simulation by assigning results, setting parameters for stimuli elements, and executing the algorithm for the given sequence.

3.0.3 Group

This module is designed to initialize the group with a name, number of phases, and model, create and clear maps, add entries to maps, and retrieve entries from maps. It also has functions for retrieving the total maximum duration and the group's name.

3.0.4 Raw stim

This module has been designed to process stimuli by loading raw data, running the CNN model, and creating filter maps.

3.0.5 Sequential

This module is designed as a neural network architecture with 5 convolutional layers and 4 max pooling layers. The forward pass of this class involves passing the input through the layers and returning the output.

3.0.6 Trial

This module is used to create and manipulate trial objects. These objects contain trial strings, cues, and trial numbers, and can be copied, modified, and reinforced. This class also includes methods for getting and setting cues, trial strings, and trial numbers. This is important for data parsing.

3.0.7 Base

This module is designed as an abstract base class that defines the common properties and methods of neural network layers. It includes methods for forward propagation, setting weights, and retrieving weights. The class also distinguishes between trainable and non-trainable layers.

3.0.8 Config

This module contains methods to check if a given name is a context or a US. The CS class has methods to initialize the values of its instances and get their names, symbols, onset, offset, alpha, and salience.

3.0.9 Element

This module is designed to be initialized with various attributes such as index, parent, group, name, std, trials, and total stimuli. It also sets up various keys and maps for the class instance.

3.0.10 ITI

The ITI class defines a waiting period of a specified duration, while the Timing Configuration class creates timing configurations for trials using cues and sets the maximum offset value for all cues in a trial. The ITI Config class sets the minimum waiting period.

3.0.11 Model

The SOARN model class includes methods for initializing and setting various parameters related to alpha rates, context, cues, and groups. These methods allow for customization and fine-tuning of the SOARN algorithm.

3.0.12 Pooling

This module performs down-sampling by dividing the input into rectangular pooling regions and outputting the maximum value of each region. The layer takes an input array and applies the max-pooling operation to it, returning an output array with reduced dimensions.

3.0.13 Stim

The Stimulus class is initialized with various attributes such as group, symbol, trials, total stimuli, and configurations. It contains methods for initializing a stimulus with its name, number of trials, cue names, reinforcement status, and raw data. It also has methods for adding parts to the stimulus, checking if it contains a certain part or cue, getting its name, parts, cue names, and number of trials, and setting its parts.

3.0.14 Utils

This module contains all the helper functions that make running the model much easier. These include preparing and loading image data, including resizing and converting to RGB format. There is also functionality for saving images as a pickle file that is later used in training.

We implement these module packages together with functions that run the operations of the computations of the model. We aim to reduce redundancy in code implementation by making use of these classes to avoid code repetition and errors. Many packages have been used to support these implementations, the main one being the NumPy array. We have avoided the use of deep learning frameworks such as TensorFlow or PyTorch, as these require preprocessing data inputs to a certain format and using a large corpus of training data. That, though, is not the aim of this implementation; hence, the reason we settled for NumPy.

Figure 3.2 provides a snapshot of the model and data pipeline we developed, but the details of the implementations are discussed in-depth in the next chapter.

Chapter 4

Computational Model Design and Implementation

4.1 Introduction

We develop a real-time associative learning computational model that simulates the learning of visual stimuli representations using associative mechanisms. The model is implemented using an elemental connectionist approach, in which the activity of elements simulates the activation process of neurons that drive learning when a stimulus is experienced.

Elements are set up as both predictors and outcomes, enabling learning to be elemental. An error correction learning framework has been implemented using a dynamic asymptote that measures the distance between two concurrent active elements and adapts the sum of the predicted values of the predicting element by reducing the error factor.

4.2 Architecture

The implementation of SOARN required solving several technical challenges. The model processes 56×56×4 receptive fields, resulting in 12,544 elements per stimulus. Managing the computational load while maintain-

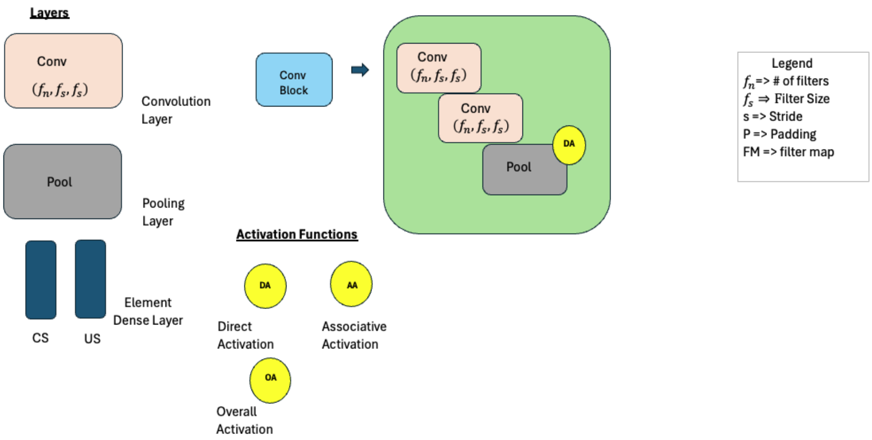


Figure 4.1: SOARN architecture components and data flow. The model processes visual stimuli through convolutional layers (Conv) and pooling layers (Pool) to extract feature maps. These feed into the associative learning network where elements form bidirectional connections. Activation functions include direct activation and associative activation, with overall activation determined by the maximum of both. The legend shows the symbols used throughout the architecture diagrams.

ing real-time performance required careful design of the activation and learning mechanisms. The model uses several configurable parameters that control learning dynamics and temporal processing. Default parameter values and their descriptions are provided in Appendix C.2.

4.2.1 Forward Propagation

We develop an input layer that reads an image of shape $(\boldsymbol{H}, \boldsymbol{W}, \boldsymbol{C})$, where $\boldsymbol{H}$ is the height of the image, $\boldsymbol{W}$ is the width and $\boldsymbol{C}$ is the channel of the image. The three-dimensional image is processed by the convolution layer which processes data by employing the convolution operation.

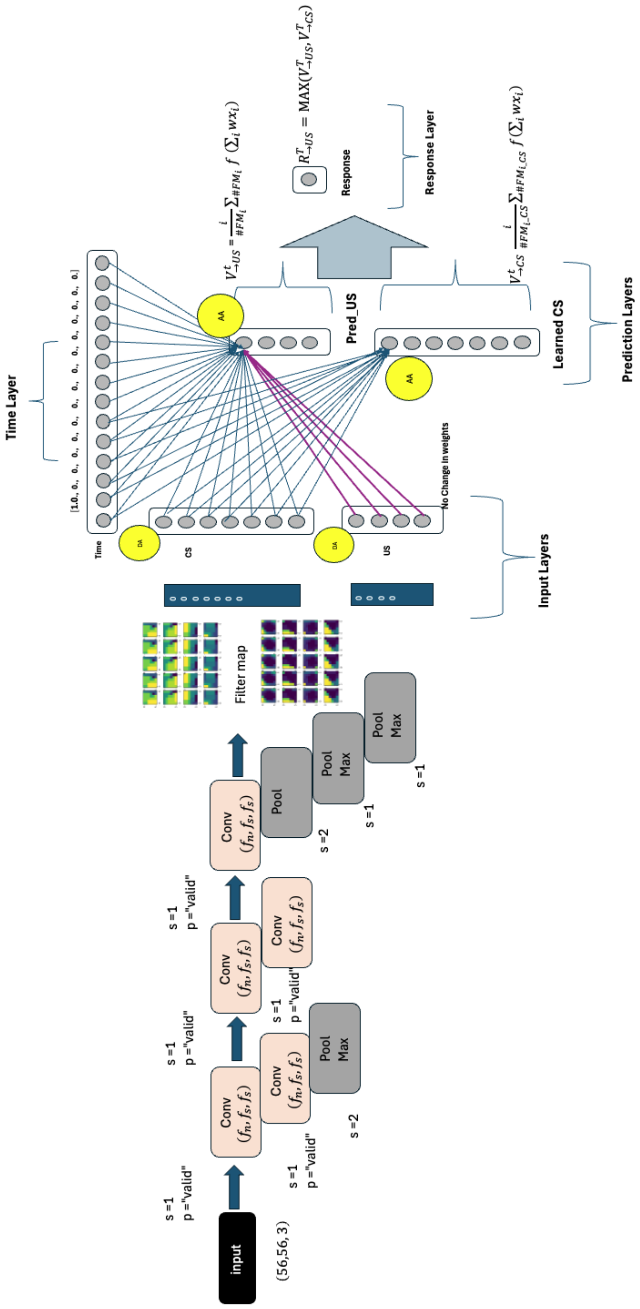


Figure 4.2: Complete SOARN architecture showing the flow from visual input to associative learning. Input images are processed through multiple convolutional layers (Conv) and pooling layers (Pool) to extract hierarchical features. The Time Layer distributes activation across temporal elements, which then feed into the associative network. Elements are fully connected both within and between layers, forming bidirectional associations. The Predicted Layers show the learned representations for each stimulus. Mathematical operations shown include convolution (equation boxes), element-wise multiplication, and summation across connections.

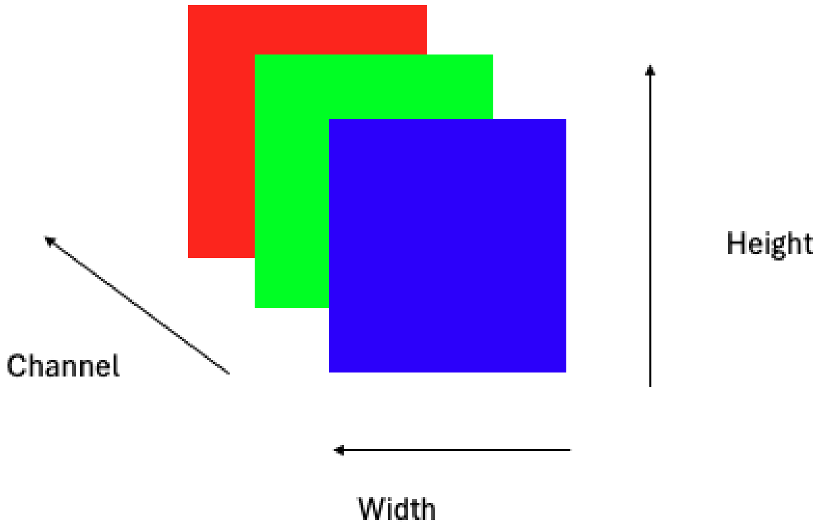


Figure 4.3: RGB channel structure of input images. Color images consist of three channels (Red, Green, Blue) with dimensions Height × Width × 3. Each channel contains the intensity values for its respective color component.

4.2.2 Convolutional Layer Computation

The convolution operation transforms input images into feature maps. The kernel slides over the whole input and on each slide, element-wise multiplication is performed followed by a summation of all values to get a single value. These output maps are stacked together to form the channels of the new image.

4.2.2.1 CNN Processing

The CNN architecture consists of five convolutional layers and four pooling layers. This five-layer design was chosen to match the hierarchical processing levels in biological vision: edge detection, texture, shape, object parts, and full objects. Each layer progressively extracts more complex features while maintaining computational efficiency. Each convolutional layer uses 4 filters with 3×3 kernels. This minimal configuration captures color plus two additional feature detectors.

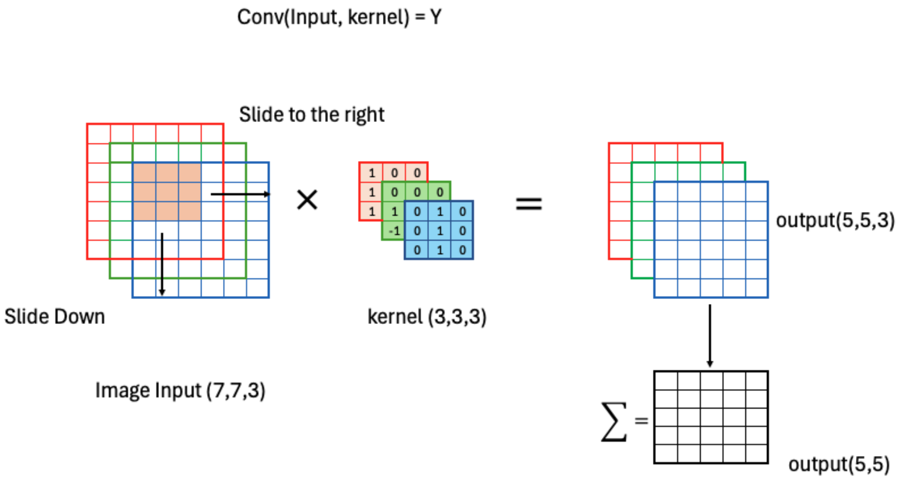
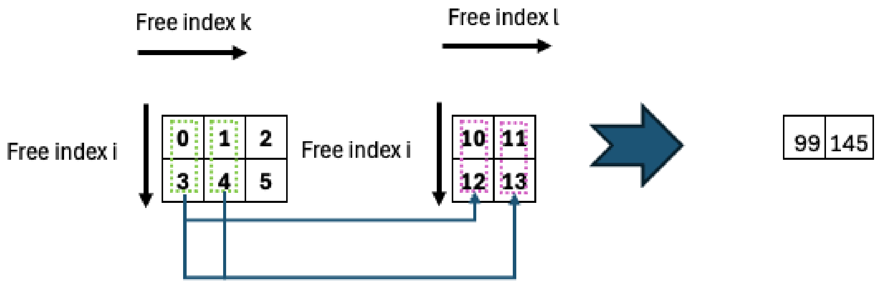


Figure 4.4: Convolution operation showing kernel sliding across input. A 3×3 kernel performs element-wise multiplication with the input image at each position, followed by summation to produce a single output value. The kernel slides across the entire input with specified stride to generate the complete output feature map.



$$99 = (0 * 10) + (1 * 11) + (3 * 12) * (4 * 13)$$

$$145 = (1 * 10) + (2 * 11) + (4 * 12) * (5 * 13)$$

Figure 4.5: Element-wise multiplication in convolution operation. Each element of the 3×3 kernel is multiplied with the corresponding element in the input image patch. The products are then summed to produce a single output value (99 + 145 = 244 in this example).

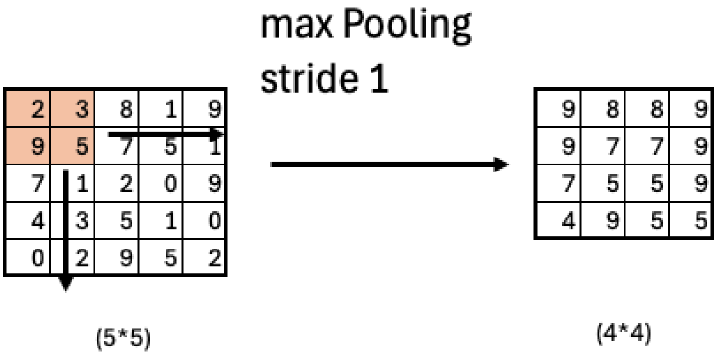
The implemented convolution operation is represented by

$$a_{ik} b_{il} = \begin{cases} a_{0k} * b_{0l} + a_{1k} * b_{1l} \\ \sum_i a_{ik} b_{il} \\ a^T[k].b^T[l] \end{cases} \tag{4.1}$$

where i is the index of the matrix and k and l are the free indices. (See Appendix A.1 for detailed implementation code and parameter specifications.)

4.2.3 Pooling Layer Computation

The pooling layer reduces the dimensionality $(H \times W)$ of the image, but not the number of channels. The operation applied is max pooling, with different strides applied on the different layers of the model.



2	3	8	1	9
9	5	7	5	1
7	1	2	0	9
4	3	5	1	0
0	2	9	5	2

9	8	8	9
9	7	7	9
7	5	5	9
4	9	5	5

Figure 4.6: Max pooling operation with stride 1. A 2×2 pooling window selects the maximum value from each region of the input (5×5) to produce the output (4×4). This operation reduces spatial dimensions while preserving the most prominent features.

The operation is executed for the entire image, resulting in a reduced feature map that contains the defined maximum values of the kernel operation. (Complete pooling implementation details are provided in Appendix A.2.)

4.3 Stimulus Representation and Activation

The reservoir of feature maps at the end of the pooling layer are transformed into a matrix of units, each belonging to the specific stimulus categories (CS or outcome). The unit's basic function is to compute the activation based on the number of inputs connected to it.

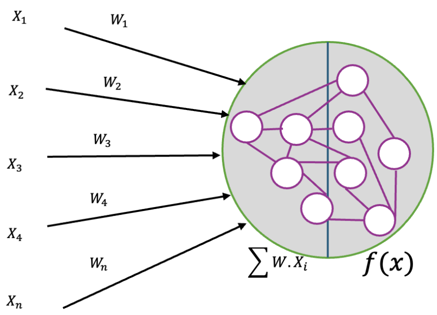


Figure 4.7: Single element as a computational unit. Each element receives multiple weighted inputs ($X_1...X_n$ with weights $W_1...W_n$), computes their weighted sum, and applies an activation function $f(x)$ to produce output. Elements serve as both predictors and outcomes in the associative network.

The predictive computation of this element takes the form of a sum of the multiplication of all the net inputs $X_1...X_n$, and their respective weights $W_1...W_n$:

$$y = \sum_{i=1}^{n} w_i \cdot x_i \tag{4.2}$$

4.3.1 Activation Overview

For each time step of the stimulus duration, there are elements e_i representing the presence of the stimulus at that point. Each stimulus contains a pre-set duration T that indicates an onset time t_{on}, offset time t_{off}, and inter-trial interval $\boldsymbol{iti}$ before the next trial onset.

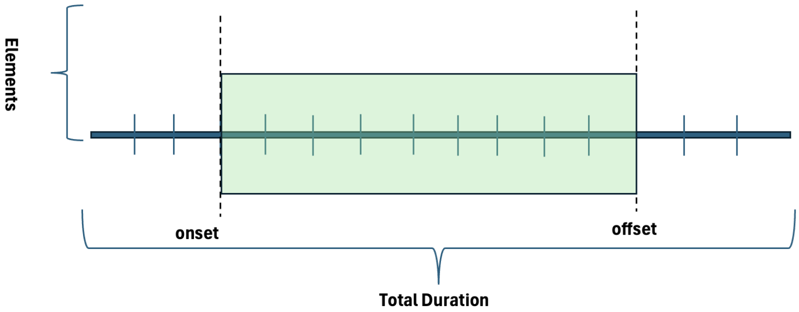


Figure 4.8: Stimulus experience timeline showing temporal parameters. Each stimulus has an onset time, offset time, and total duration. The inter-trial interval separates consecutive trial presentations. Elements are activated during the stimulus duration.

These elements are activated at each time step by a modified normal distribution activation function:

$$\boldsymbol{A}_i^t = \exp\left(-\frac{(\boldsymbol{t}-\boldsymbol{i})^2\boldsymbol{k}}{2\boldsymbol{\sigma}_i^2}\right) \times \boldsymbol{I} \tag{4.3}$$

where $\boldsymbol{\sigma}_i^2 = \boldsymbol{i}\cdot\boldsymbol{\delta}^2$, ensuring that elements with higher indices have broader temporal receptive fields (see Appendix A.3 for activation profiles).

4.3.1.1 Direct Activation Operation

We represent the life cycle of each sub-element of the constituent cluster using a Markov property, where each element is represented by two states: an inactive state $\boldsymbol{I}$ and an active state $\boldsymbol{A}$.

$$\boldsymbol{e}_i = \begin{cases} \boldsymbol{S}_1 = \boldsymbol{I} & \text{inactive} \\ \boldsymbol{S}_2 = \boldsymbol{A} & \text{active} \end{cases} \tag{4.4}$$

From the Markov chain property, each element builds its own transition matrix populated by probability values of moving from one state to another at every time step.

$$\begin{array}{ccc} & \mathbf{I} & \mathbf{A} \\ \mathbf{I} & 1 - P_I & P_I \\ \mathbf{A} & P_A & 1 - P_A \end{array}$$

Figure 4.9: Transition matrix for element state changes. Each element can be in either an inactive state (I) or active state (A). The matrix shows transition probabilities between states, where $\boldsymbol{P_I}$ represents the probability of remaining inactive (1 - $\boldsymbol{P_A}$) and $\boldsymbol{P_A}$ represents the probability of transitioning to or remaining in the active state. These probabilities govern the stochastic activation of sub-elements during stimulus presentation.

4.3.1.2 Associative Activation

Model elements are connected by weight values $\boldsymbol{W_{ij}}$ to one another. Each element can retrieve activity from elements connected to it when serving as a predictor. The associative activation is:

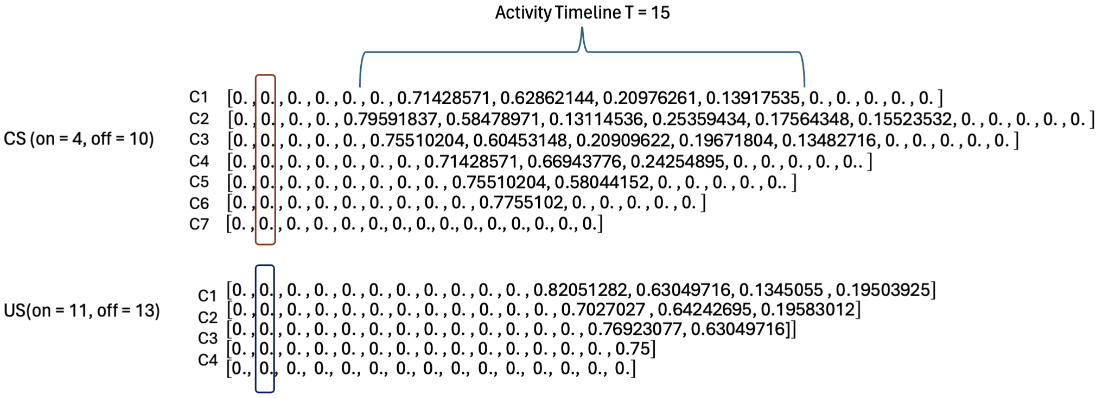
$$\boldsymbol{AA_i^t} = \sum_j \boldsymbol{A_j^t} \times \boldsymbol{W_{ji}} \tag{4.5}$$

The overall activity of the element compares the direct and associative activities:

$$\boldsymbol{OA_i^t} = \max(\boldsymbol{A_i^t}, \boldsymbol{AA_i^t}) \tag{4.6}$$

4.4 Network

The network object is an engine that applies all the operations we have designed to learn experiments.The full network contains over 150,000 modifiable connections for a typical three-stimulus experiment. To handle this scale, we implemented a sparse activation scheme where only elements with $OA_i^t > 0.01$ participate in weight updates, reducing the computation cost without affecting learning outcomes. The complete network object structure and implementation details are described in Appendix C.3.



```
C1  [0. , 0. , 0. , 0. , 0. , 0. , 0.71428571, 0.62862144, 0.20976261, 0.13917535, 0. , 0. , 0. , 0. , 0. ]
C2  [0. , 0. , 0. , 0. , 0.79591837, 0.58478971, 0.13114536, 0.25359434, 0.17564348, 0.15523532, 0. , 0. , 0. , 0. , 0. ]
C3  [0. , 0. , 0. , 0. , 0. , 0.75510204, 0.60453148, 0.20909622, 0.19671804, 0.13482716, 0. , 0. , 0. , 0. , 0. ]
C4  [0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.71428571, 0.66943776, 0.24254895, 0. , 0. , 0. , 0. , 0.. ]
C5  [0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.75510204, 0.58044152, 0. , 0. , 0. , 0. , 0.. ]
C6  [0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.7755102, 0. , 0. , 0. , 0. , 0. ]
C7  [0. , 0. , 0. , 0. , 0. , 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]

C1  [0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.82051282, 0.63049716, 0.1345055 , 0.19503925]
C2  [0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.7027027 , 0.64242695, 0.19583012]
C3  [0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.76923077, 0.63049716]]
C4  [0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.75]
    [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]
```

Figure 4.10: Input activity patterns across time showing element activation matrices. Each row represents a time step with activity values for CS and US elements. The matrices display binary activation patterns (0 or 1) for different stimulus combinations across the trial duration, illustrating how stimuli are represented temporally in the network.

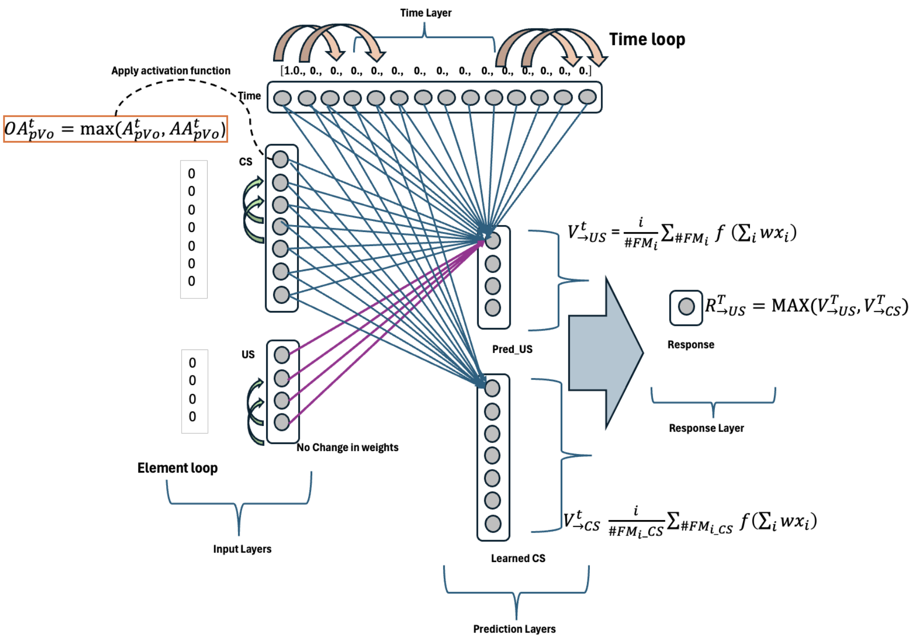
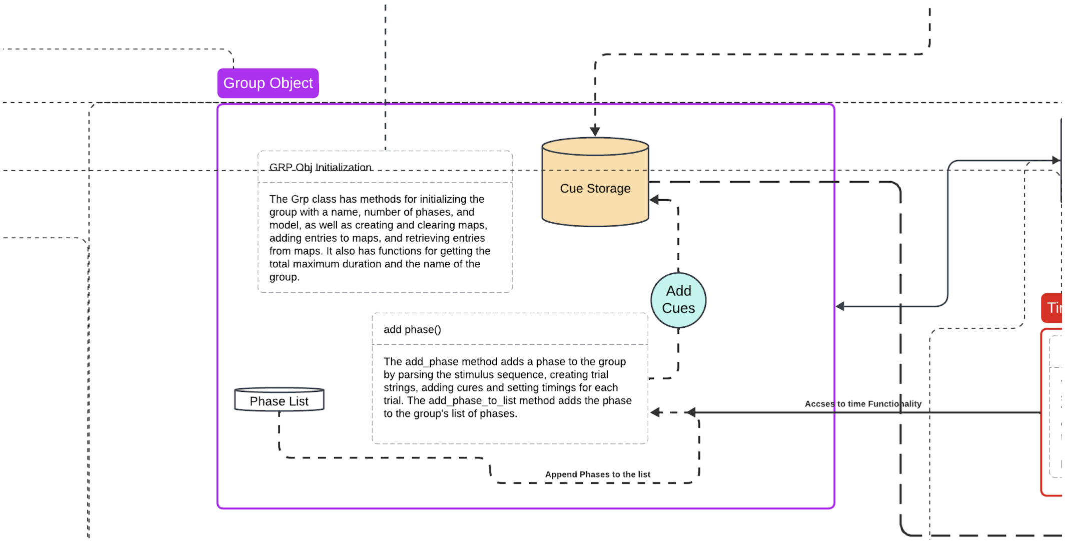

Time Layer
Time loop
Apply activation function
[1.0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]
Time
$OA_{pVo}^{t} = \max(A_{pVo}^{t}, AA_{pVo}^{t})$
CS
0
0
0
0
0
0
0
$V_{\to US}^{t} = \frac{i}{\#FM_i} \sum_{\#FM_i} f\ (\sum_i wx_i)$
$R_{\to US}^{T} = \mathrm{MAX}(V_{\to US}^{T}, V_{\to CS}^{T})$
Response
Pred_US
US
0
0
0
0
No Change in weights
Response Layer
Element loop
$V_{\to CS}^{t}\ \frac{i}{\#FM_{i\_CS}} \sum_{\#FM_{i\_CS}} f(\sum_i wx_i)$
Input Layers
Learned CS
Prediction Layers

Figure 4.11: Network structure showing bidirectional connectivity between layers. Input layers process time-tagged stimuli which connect to element layers through weighted connections. Each element receives both direct activation (from sensory input) and associative activation (from connected elements). The network computes overall activation as the maximum of direct and associative inputs, with predicted layers showing the learned representations.

4.4.1 Group and Phase Objects

The group object performs dynamic tasks that enable the model to perform several operations based on the specified tests and experiments. The phase object on the other hand, is central to running simulations, setting up simulation parameters and trial sequences.

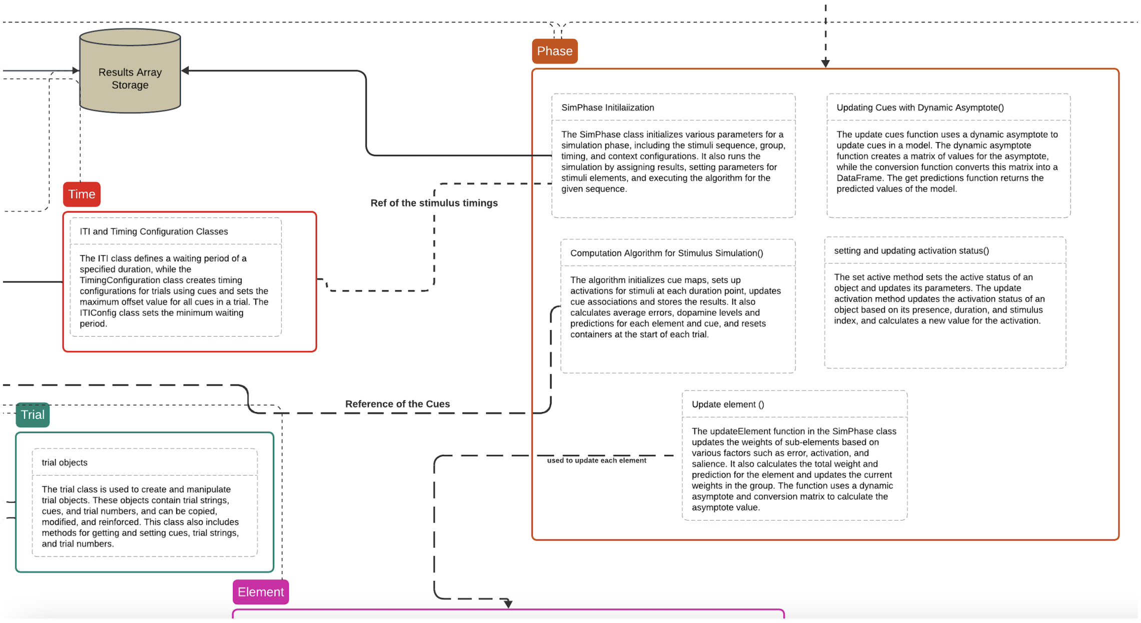


The Grp class has methods for initializing the group with a name, number of phases, and model, as well as creating and clearing maps, adding entries to maps, and retrieving entries from maps. It also has functions for getting the total maximum duration and the name of the group.

The add_phase method adds a phase to the group by parsing the stimulus sequence, creating trial strings, adding cures and setting timings for each trial. The add_phase_to_list method adds the phase to the group's list of phases.

Figure 4.12: Group object architecture and functionality. The group object manages experimental groups through initialization methods that set up group name, number of phases, and model parameters. It maintains a cue storage database and phase list, with methods to add phases by parsing stimulus sequences, creating trial strings, adding cues, and setting timings for each trial. The object provides access to time functionality and appends phases to enable flexible experimental design across multiple training phases.

Both Group and phase objects implement a flexible experimental protocol system. A single group can run multiple phases with different stimulus configurations including handling trial randomization and timing. For instance, a blocking experiment with 100 A+ trials followed by 100 AB+ trials is specified as: `group.add_phase('100A+/100AB+')` The parser automatically generates the trial sequence and timing parameters (see Appendix C.3 for parser implementation).


Results Array Storage
Phase
Time
Ref of the stimulus timings
ITI and Timing Configuration Classes
SimPhase Initilaiization
Updating Cues with Dynamic Asymptote()
Computation Algorithm for Stimulus Simulation()
setting and updating activation status()
Reference of the Cues
Trial
trial objects
Update element ()
used to update each element
Element

The ITI class defines a waiting period of a specified duration, while the TimingConfiguration class creates timing configurations for trials using cues and sets the maximum offset value for all cues in a trial. The ITIConfig class sets the minimum waiting period.

The SimPhase class initializes various parameters for a simulation phase, including the stimuli sequence, group, timing, and context configurations. It also runs the simulation by assigning results, setting parameters for stimuli elements, and executing the algorithm for the given sequence.

The update cues function uses a dynamic asymptote to update cues in a model. The dynamic asymptote function creates a matrix of values for the asymptote, while the conversion function converts this matrix into a DataFrame. The get predictions function returns the predicted values of the model.

The algorithm initializes cue maps, sets up activations for stimuli at each duration point, updates cue associations and stores the results. It also calculates average errors, dopamine levels and predictions for each element and cue, and resets containers at the start of each trial.

The set active method sets the active status of an object and updates its parameters. The update activation method updates the activation status of an object based on its presence, duration, and stimulus index, and calculates a new value for the activation.

The trial class is used to create and manipulate trial objects. These objects contain trial strings, cues, and trial numbers, and can be copied, modified, and reinforced. This class also includes methods for getting and setting cues, trial strings, and trial numbers.

The updateElement function in the SimPhase class updates the weights of sub-elements based on various factors such as error, activation, and salience. It also calculates the total weight and prediction for the element and updates the current weights in the group. The function uses a dynamic asymptote and conversion matrix to calculate the asymptote value.

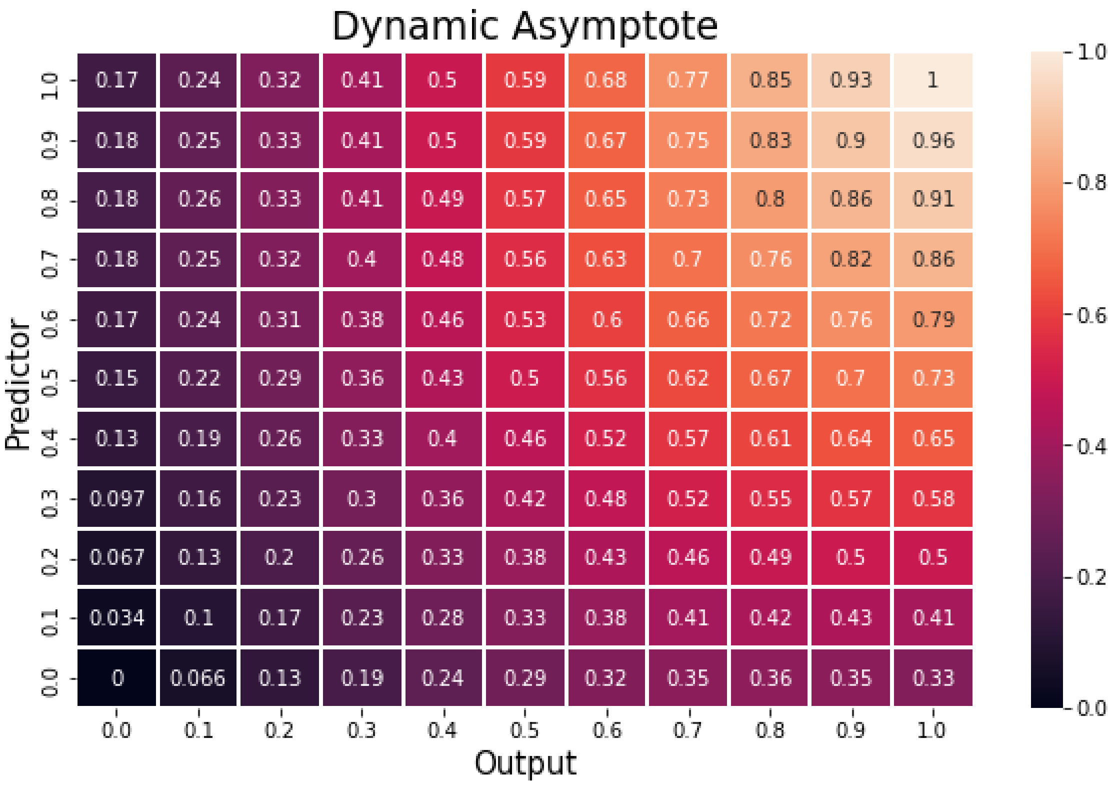
Figure 4.13: Phase object structure and workflow. The phase object coordinates the simulation process, managing the flow from trial configuration through stimulus timing to result storage. It initializes simulation parameters, updates cues with dynamic asymptotes, and executes the computation algorithm for stimulus simulations. The object interfaces with timing configurations, trial objects, and the element update system to orchestrate the complete learning process across specified trial sequences.

4.5 Learning Mechanism

4.5.1 Dynamic Asymptote

The dynamic asymptote of learning, $\boldsymbol{\lambda}$, measures the degree to which elements vary in activity:

$$\boldsymbol{\lambda} = \left(\boldsymbol{A}_{\max} - \frac{(\boldsymbol{OA}_p^t - \boldsymbol{OA}_o^t)^2}{\mathbf{2}} \right) \times \left(\boldsymbol{\nu} \cdot \boldsymbol{OA}_o^t + (\mathbf{1} - \boldsymbol{\nu}) \cdot \boldsymbol{OA}_p^t \right) \tag{4.7}$$



	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
1.0	0.17	0.24	0.32	0.41	0.5	0.59	0.68	0.77	0.85	0.93	1
0.9	0.18	0.25	0.33	0.41	0.5	0.59	0.67	0.75	0.83	0.9	0.96
0.8	0.18	0.26	0.33	0.41	0.49	0.57	0.65	0.73	0.8	0.86	0.91
0.7	0.18	0.25	0.32	0.4	0.48	0.56	0.63	0.7	0.76	0.82	0.86
0.6	0.17	0.24	0.31	0.38	0.46	0.53	0.6	0.66	0.72	0.76	0.79
0.5	0.15	0.22	0.29	0.36	0.43	0.5	0.56	0.62	0.67	0.7	0.73
0.4	0.13	0.19	0.26	0.33	0.4	0.46	0.52	0.57	0.61	0.64	0.65
0.3	0.097	0.16	0.23	0.3	0.36	0.42	0.48	0.52	0.55	0.57	0.58
0.2	0.067	0.13	0.2	0.26	0.33	0.38	0.43	0.46	0.49	0.5	0.5
0.1	0.034	0.1	0.17	0.23	0.28	0.33	0.38	0.41	0.42	0.43	0.41
0.0	0	0.066	0.13	0.19	0.24	0.29	0.32	0.35	0.36	0.35	0.33

Figure 4.14: Dynamic asymptote lookup table showing activity-dependent learning ceiling values. The asymptote varies as a function of predictor activity (rows) and outcome activity (columns), ranging from 0.0 to 1.0. Higher values (red) occur when predictor and outcome activities are similar, while lower values (blue) occur when activities differ. This mechanism ensures that learning is strongest when elements have comparable activation levels, implementing the principle that co-active elements form stronger associations.

This formulation solves a key limitation of static learning rates. When $OA_p^t = OA_o^t$, λ reaches its maximum, promoting learning between co-active elements. Conversely, when activities differ maximally, λ approaches zero, preventing spurious associations. Implementing the lookup table 4.14 allows efficient computation during the thousands of weight updates per trial (see Appendix A.4 for computational complexity analysis).

4.5.2 Weight Update

Learning involves changing weight components through error correction:

$$\boldsymbol{W_{ij}^{t+1} = \Delta W_{ij}^{t} + W_{ij}^{t}} \tag{4.8}$$

where:

$$\boldsymbol{\Delta W_{ij}^{t} = (\lambda_j^t - V_j^t) \times (\lambda_i^t - V_i^t) \times S_i \times S_j \times b} \tag{4.9}$$

The backward discount value $\boldsymbol{b}$ only applies if the predictor occurs after the outcome in time.

4.5.3 Response

At the end of a trial, the model response is:

$$\boldsymbol{R^T = \max(V_{\rightarrow \text{US}}^{T}, V_{\rightarrow \text{CS}}^{T})} \tag{4.10}$$

where:

$$\boldsymbol{V_{\rightarrow A}^{T} = \frac{1}{FM_{(m\times n)}} \sum_{\forall FM_i} f\left(\sum_i w_i x_i\right)} \tag{4.11}$$

4.6 Summary

This implementation successfully processes 256×256 RGB images through associative learning, a significant advancement over traditional models limited to abstract representations. The model runs a 200-trial experiment in approximately 30-45 minutes on standard hardware, extracting both associative measures (V values) and novel receptive field visualizations. The modular architecture allows us to test new learning rules by modifying only the weight update function (4.9), while the visualization capabilities provide insight into how associations modify perceptual representations.

Chapter 5

Model Experiments and Results

5.1 Introduction to Experiments

Please note that in the following experiments, the notation + is used to match the experimental designs conducted in the laboratory. However, in all subsequent experiments, the outcome is a neutral stimulus. The objective is not to measure the change in response due to the formation of a predictor-outcome association but to assess the role of the associative process in the formation of more complex compound representations between two or more constituents. The analysis, however, is presented in predictive terms.

5.2 Evaluation Framework for Associative Learning

5.2.1 Challenges in Evaluating Associative Learning Models

The evaluation of associative learning models differs fundamentally from supervised learning where performance can be measured against labelled data. Unlike supervised learning with its predefined correct outputs, associative learning models must demonstrate their validity by reproducing a range of well-established phenomena observed in animal learning studies.

The field has evolved significantly from early verbal theories to mathematical models that make precise, testable predictions. For example, temporal difference learning models now link behavioral phenomena such as blocking to specific neural mechanisms dopamine prediction errors, while models like those reviewed by [6] provide unified mathematical formalisms to compare different theoretical approaches.

One fundamental challenge stems from the theoretical divide within the field. As highlighted by Shanks [168], there is an ongoing tension between association-based theories that employ concepts such as excitation and inhibition, and cognitive theories that invoke hypothesis testing and explicit reasoning about causal relationships. This divide complicates the evaluation framework approach because the same phenomena can be explained through different theoretical mechanisms.

Furthermore, even within associative frameworks, models differ in their core assumptions. Some focus on changes in stimulus processing, that is, CS associability, while others emphasize prediction error mechanisms, that is, US processing [60]. This diversity of approaches adds a layer of complexity to model comparison and evaluation.

5.2.2 Our Evaluation Approach

Given these challenges, we evaluate our model using multiple criteria. We test whether the model reproduces fundamental phenomena from the conditioning literature: acquisition, extinction, blocking [18], conditioned inhibition [12, 17], and negative patterning. For each phenomenon, we examine both the qualitative pattern and the quantitative trajectory of learning.

Through visualization of receptive fields, we provide direct evidence of how associative processes modify perceptual representations. This capability allows us to evaluate not only whether the model produces correct outputs but also the mechanisms through which these outputs are generated.

Critically, all phenomena are tested using fixed parameter set. This means that the same learning rate, salience values, and model parameters are used in all experiments, from simple acquisition to complex negative pattern discriminations. This constraint ensures that the model's performance reflects theoretical coherence rather than adjusting parameters after seeing results to make each phenomenon work.

5.2.3 Metrics Used in This Work

Following the Rescorla-Wagner [11], our primary metric is associative strength (V), representing the predictive value each stimulus acquires through learning. We track V values across trials to generate learning curves that reveal the pattern of the learning process.

Beyond these traditional measures, our model uniquely provides access to internal representations through receptive field visualization. Although conventional models report that a stimulus has acquired a certain V value, our approach reveals how the stimulus representation itself has changed providing visual insight into the mechanisms underlying the associative phenomena.

The combination of associative strength values and a visualization of the receptive fields allows for a more complete evaluation of the model's performance. We can assess not only whether the model reproduces known phenomena through appropriate V values but also whether it does so through biologically plausible mechanisms that align with our understanding of perceptual and learning systems.

5.2.4 What are Receptive Fields?

In our model, receptive fields are the visual feature patterns extracted by the CNN layers. Each receptive field represents what a particular unit in the network "sees" or responds to when processing an image. These are visualized as the 56×56 feature maps output by our CNN architecture, where different patterns and intensities indicate which visual features have been detected and learned.

5.2.5 Extracting and Visualizing Receptive Fields

Our model extracts receptive fields from the final pooling layer of the CNN, producing feature maps of size (5, 56, 56) where:

- 5 represents the number of feature maps
- 56×56 represents the spatial resolution

These receptive fields capture what the network sees when presented with a stimulus. They represent the learned internal representation of the visual input after processing through the hierarchical CNN layers.

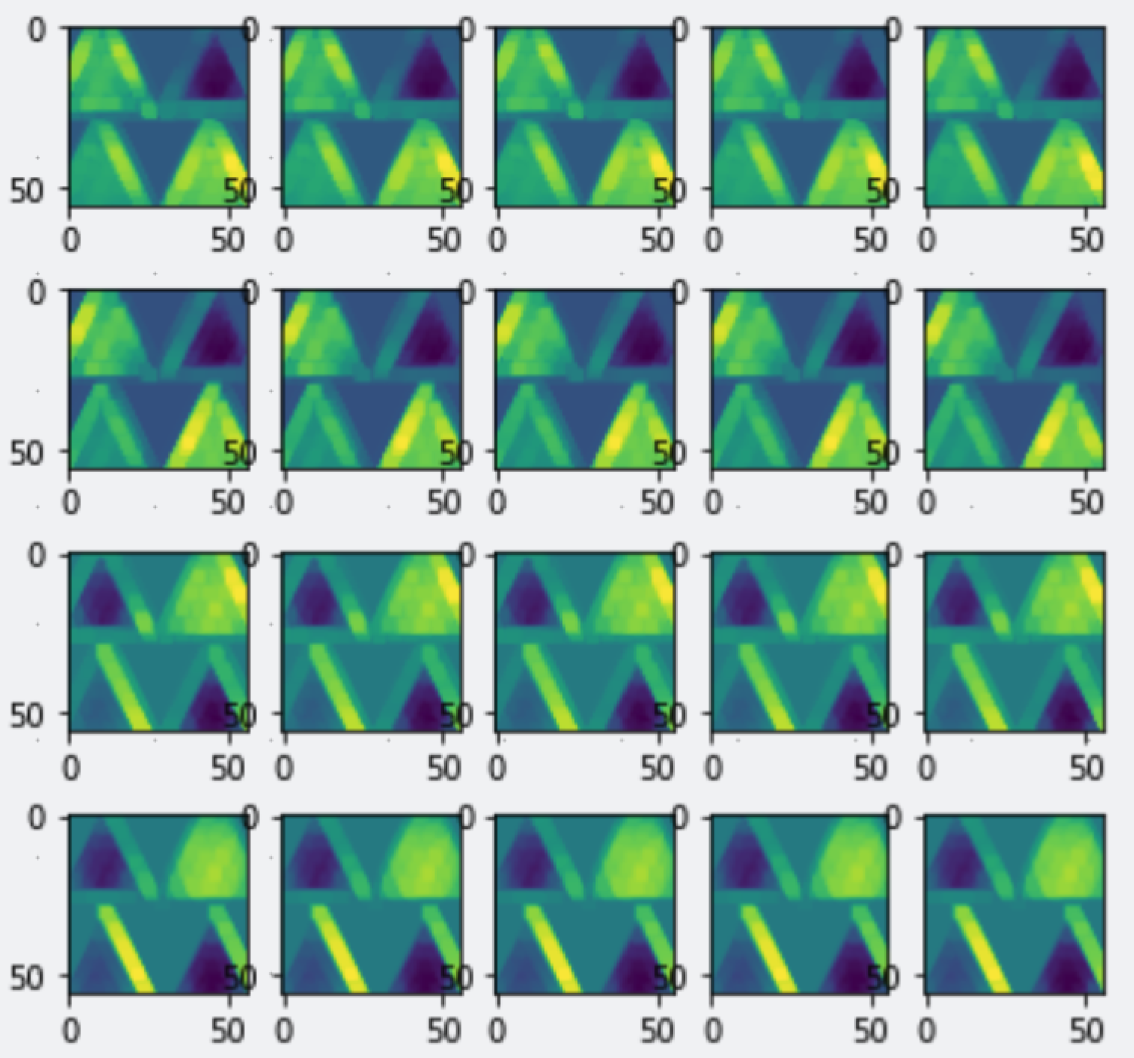


Figure 5.1: Example of CNN receptive fields before associative learning. This 4×5 grid shows 20 different receptive fields (each 56×56 pixels) extracted from the CNN's final pooling layer. The colour patterns (purple, green, yellow) represent different feature activations. Notice the clear geometric patterns - these are the visual features the CNN extracts from a triangle stimulus before any learning occurs.

The transformation between these two states reveals how associative learning shapes perception. Before learning (Figure 5.1), the receptive fields show clear triangular patterns. After learning (Figure 5.2), circular features from the outcome have been incorporated, creating a complex representation structure.

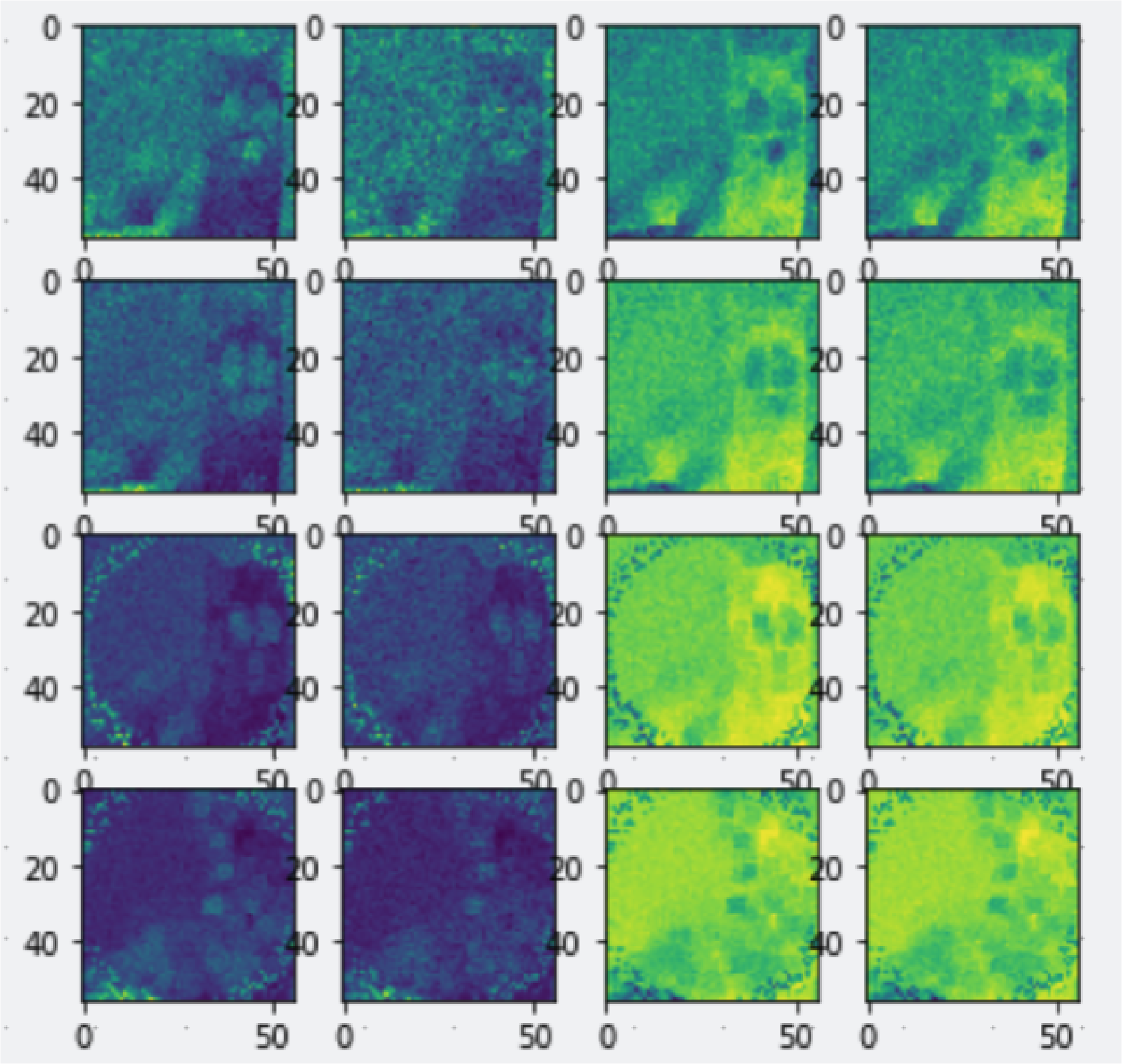


Figure 5.2: Receptive fields after associative learning shown across different time points within a trial. Each column represents a different time step, demonstrating the temporal evolution of the representation. The progression from left to right shows how the features change over the course of stimulus presentation, with circular patterns (bright yellow/-green regions) becoming more prominent in later time steps. This reveals how associative learning modifies perceptual representations during stimulus processing.

5.3 Acquisition

5.3.1 Experimental Design

The experimental design we use in this model tests whether the simultaneous pairing of two stimuli, a $CS(A)$ and an outcome $+$ would endow the former with a predictive capability of the latter, incorporating its characteristic features. In other words, whether the model would allow the acquisition of a relationship $CS — +$ that would modify the input predictor image into a new representation that incorporates some of the features of the predicted outcome. Training in the experiment involved a single phase of 100 trials, in which stimulus A (a display of four triangles of different colours arranged in a square shape) was presented, followed by an outcome (a large empty circle). Following training, we tested how much of the CS was able to predict the outcome, reproducing its image in a compound with the CS. This was examined by generating a receptive field map.

Table 5.1 is the CNN layer setup. The two images, **A** and **+**, were passed through the convolution and pooling layers, followed by the running of the trials specified in Table 5.2. The base image size is (256*256), and the output size after CNN is (4, 56, 56), where 4 is the number of feature maps.

5.3.2 Results and Evaluation

The acquisition test measured the level of prediction that image A conveyed of the outcome features. Similarly to the Rescorla-Wagner model,

CNN set-up				
Layers	**Filters**	**Kernel Size**	**Padding**	**Stride**
Conv2D	4	3x3	'valid'	1
Conv2D	4	3x3	'valid'	1
POOL$_{max}$	4	4x4		2
Conv2D	4	3x3	'valid'	1
Conv2D	4	3x3	'valid'	1
Conv2D	4	3x3	'valid'	1
POOL$_{max}$	4	4x4		2
POOL$_{max}$	4	4x4		1
POOL$_{max}$	4	4x4		1

Table 5.1: Convolution and pooling stack used in all experiments. This architecture progressively reduces spatial dimensions while maintaining 4 feature channels throughout.

Group	Phase 1	Test
1	100A+	A?

Table 5.2: Acquisition Design. 100 trials of stimulus A paired with outcome +, followed by test of A alone.

[11], in our model, learning is driven by the discrepancy between the actual outcome and its prediction. While a discrepancy exists between the predicted value and the outcome, learning takes place. As the prediction value increases, learning decreases. The acquisition pattern does follow a negative accelerated curve semi-asymptotic around the maximal prediction value.

Figure 5.3 shows the growth of the associative strength over the number of trials. In the early conditioning trials, the outcome prediction error is highest, generating larger jumps in prediction compared to subsequent trials in which the size of the increments decreases as the outcome becomes predictable and the learning approaches an asymptote. This pattern validates that our model implements error-driven learning consistent with established associative learning principles.

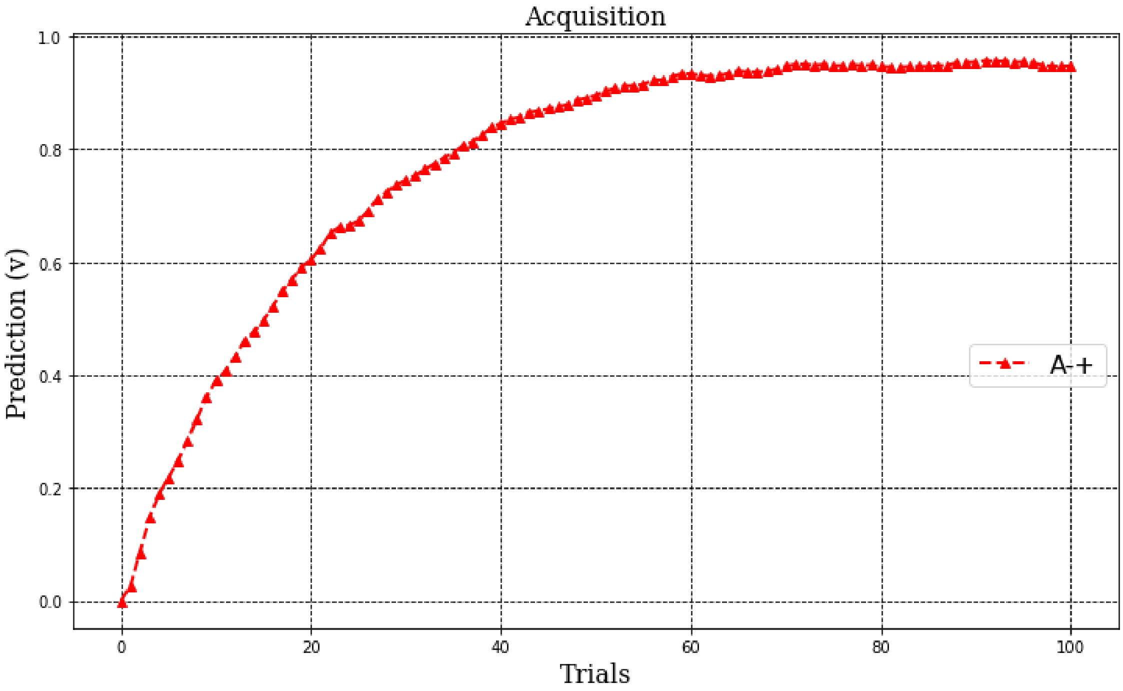


Figure 5.3: Acquisition learning curve showing growth of associative strength (V) over 100 trials. The curve demonstrates classic negatively accelerated learning, reaching asymptote around V=0.93. Early trials show rapid learning due to large prediction errors, while later trials show minimal change as the outcome becomes well-predicted. The red-dashed line indicates the asymptotic value approached by the model.

5.3.3 Receptive Field Analysis

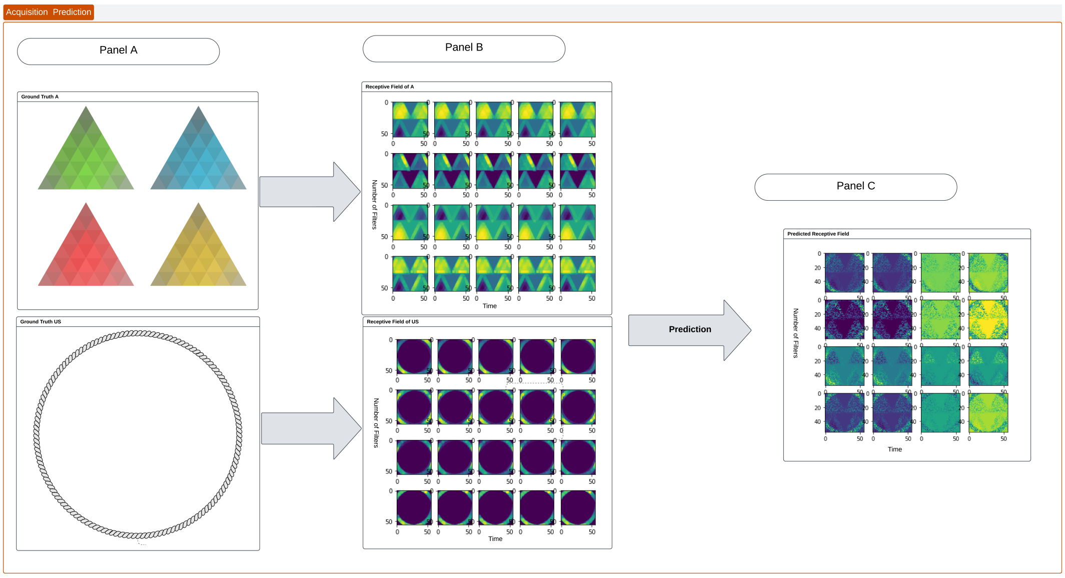


Figure 5.4: Receptive field analysis for acquisition. Panel A shows the original CS (four colored triangles) and outcome (circle) stimuli. Panel B displays the receptive fields extracted by CNN before training, showing clear triangular patterns for the CS and minimal activation for the outcome. Panel C shows the critical result: after 100 acquisition trials, the predicted receptive field demonstrates clear integration of features. The temporal progression (columns 1-4) reveals how circular features from the outcome become incorporated into the CS representation, with strongest integration visible in later time points (rightmost columns).

Figure 5.4 panel C shows the receptive field maps for the predicted feature of the predicting stimulus A and the outcome extracted at the beginning of the conditioning training by CNN, characterizing the representation of individual stimuli.

To interpret these receptive fields, we examine several key aspects. After learning, circular patterns emerge in the CS receptive fields. Panel B shows the CS fields with only triangular features matching the input stimulus. However, Panel C reveals that after 100 acquisition trials, circular features from the outcome have been integrated into the CS representation. This integration is most visible in the later time points, where bright yellow-green circular patterns appear.

The receptive fields show neural activation patterns during a test trial in which the CS is presented alone without US to probe the learned representations. The visualization captures four time points during this test presentation. Early in the trial (leftmost column), dark purple activation indicates minimal feature detection. As the CS presentation continues, activation increases and circular features become more prominent, showing that the learned associative representation unfolds dynamically during CS processing even in the absence of the US.

Despite incorporating outcome features, the CS representation maintains aspects of its original triangular structure, visible as diagonal patterns in the heatmaps. This shows that associative learning modifies rather than replaces existing representations.

A comparative visual inspection of the receptive field maps of the initial inputs with the receptive field in Panel C shows a change in the representation of A. After effective conditioning, the receptive field obtained following the presentation of A now incorporates some elements of the outcome, suggesting the formation of a more complex stimulus representation resulting from A predicting the outcome.

Delayed Conditioning Time was also considered in the acquisition experiments. The same design experiment in Table 5.2 was carried out, but inter-stimulus interval time was altered to investigate the sensitivity of conditioning to time. We presented the CS and the outcome in a delayed procedure where $\boldsymbol{A}$ had a duration of experience that was 4 seconds long with an onset of $\mathbf{0}$ and an offset of $\mathbf{4}$. On the other hand, the $+$ had a duration of $\mathbf{2}$ seconds with exposure starting at $\mathbf{2}$ and an offset of $\mathbf{4}$. The CS still signaled the US but for a shorter exposure time.

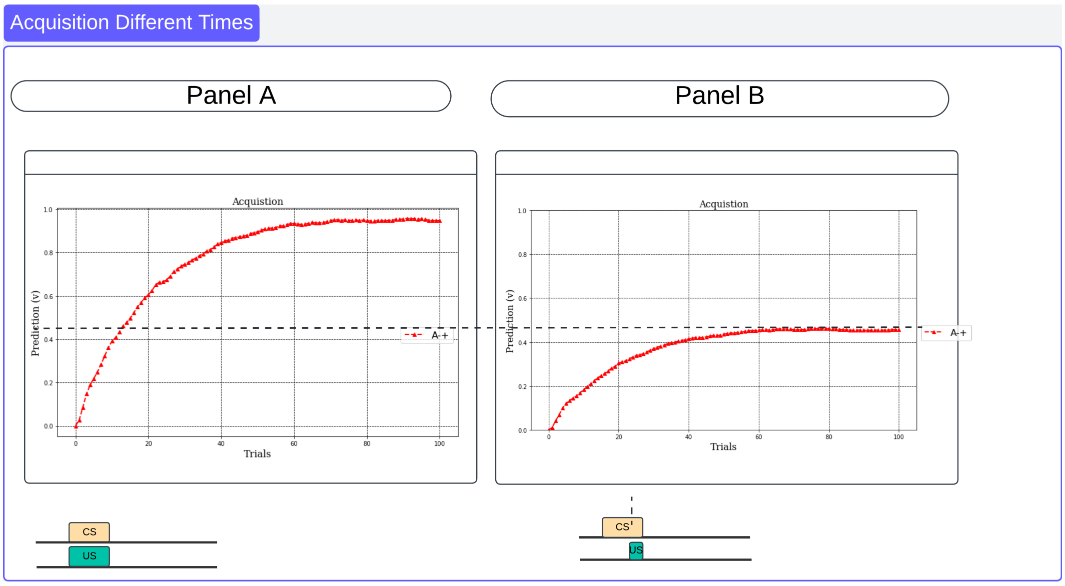


Figure 5.5: Delayed conditioning results. Panel A shows simultaneous conditioning where CS and outcome are presented together at the same time, achieving $\boldsymbol{V} \sim \mathbf{0.93}$. The timeline bars at bottom show how both stimuli are presented for the full duration (0-4 seconds), completely overlapping in time. Panel B shows delayed conditioning where the CS (top bar) is presented for the full 4 seconds while the outcome (bottom bar) appears only during the last 2 seconds of the CS presentation (from 2-4 seconds). This temporal arrangement results in reduced associative strength ($\boldsymbol{V} \sim \mathbf{0.43}$), demonstrating the model's sensitivity to temporal parameters. The difference in learning curves highlights how temporal contiguity affects association formation.

The results Figure 5.5 panel B show that there is still conditioning, although the level of conditioning decreases when the stimuli are not presented simultaneously and the exposure time of the outcome is reduced. The learning curve in delayed conditioning shows a similar negatively accelerated pattern but reaches a lower asymptote, indicating weaker association formation.

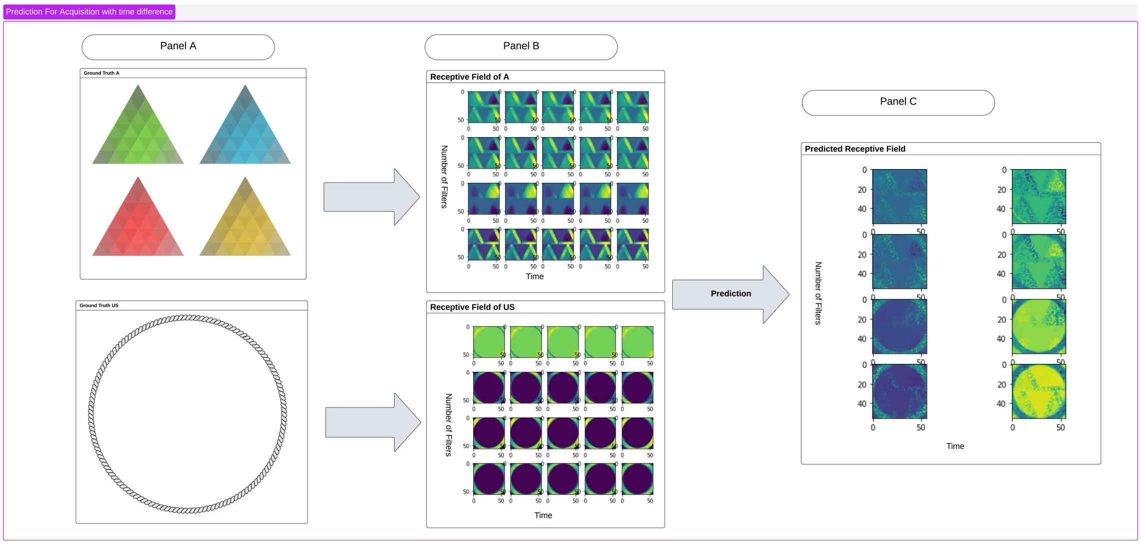


Figure 5.6: Receptive field analysis for delayed conditioning. Panel C reveals important temporal effects on representation formation. While circular features are still incorporated into the CS representation, they appear less prominent compared to simultaneous conditioning. The receptive fields shown represent the later time points where CS-outcome association occurred (when both stimuli were present together). Despite temporal overlap in the final 2 seconds, the integration of circular outcome features is weaker than in simultaneous conditioning. This pattern demonstrates that associative mechanisms can bridge temporal gaps but are less effective, resulting in weaker and less complete integration of outcome features.

Figure 5.6 panel C shows the receptive field maps for the predicted feature of the predicting stimulus A and the outcome extracted after the conditioning training by CNN.

When comparing delayed and simultaneous conditioning receptive fields, several differences emerge. The circular patterns from the outcome are less pronounced in the delayed conditioning, particularly in the number of fields showing clear circular features. This corresponds to the lower V value (0.43 vs 0.93). The delayed presentation appears to affect primarily the early processing time points, with later time points showing somewhat better integration, suggesting that the model learns temporal relationships between stimuli. The triangular features remain more dominant in delayed conditioning, indicating less modification of the original representation when temporal contiguity is reduced.

This map is not significantly different from the one observed for simultaneous conditioning, but the predicted compound representation appears to have less of the outcome features incorporated, corresponding to a lower level of V in comparison to the simultaneous experiment. These results suggest that, according to the model, the formation of a complex representation may be sensitive to time. Delayed conditioning would be less effective in generating compound representations between two stimuli than simultaneous conditioning.

5.3.4 Discussion

The acquisition results confirm that the model successfully implements fundamental associative learning. The negatively accelerated learning curve reaching asymptote around V=0.93 matches the typical pattern observed in animal conditioning studies. More significantly, the receptive field analysis reveals the mechanism underlying this learning: circular outcome features become progressively integrated into the CS representation. This visual evidence demonstrates that associative learning in our model operates by modifying perceptual representations themselves, not merely creating abstract links. These acquisition results establish the baseline against which more complex phenomena can be evaluated.

5.4 Extinction

5.4.1 Experimental Design

The experimental design we used to test for extinction is similar to the acquisition design, but the main difference is that, following an acquisition phase of 100 trials, we presented the $\boldsymbol{CS(A)}$ with no $\boldsymbol{outcome(-)}$ for another 100 trials. This experiment tests for the extinction phenomenon, which describes the progressive loss of associative strength between previously conditioned stimuli observed when the predictor is presented in isolation. We tested the performance of this model by checking whether the CS would be able to invoke the outcome representation after the extinction phase. The images used in this experiment have a base of (256*256) and a receptive field of (4,56,56). We set up an experiment containing a single group with two phases as described in Table 5.3.

Group	Phase 1	Phase 2	Test
1	100A+	100A-	A?

Table 5.3: Acquisition and Extinction Design. Phase 1 establishes the CS-outcome association, Phase 2 presents CS alone to test extinction.

In phase one, a CS A - an image display containing four triangles - was paired with a US, a circle image, the outcome. In phase two, the CS is presented without outcome $CS \rightarrow -$. A total of 100 training epochs were run in each phase. A single test to A was carried out at the end to produce a receptive field map.

5.4.2 Results and Evaluation

During conditioning in the first phase, the triangle acquires a positive value of V, as described earlier in the acquisition experiment. In the second phase, the absence of the outcome is represented with the symbol $'-'$. The computational procedure involved inputting the image with zero intensity, meaning it did not engender activity in the network. As per model definitions, such input should generate low, near zero, dynamic asymptote values, which combined with the high prediction value of the active predicting cue, results in a large negative δ error. Hence, the prediction V decreases with each extinction trial. The results indicate that extinction is not complete and proceeded more slowly than acquisition.

5.4.3 Receptive Field Analysis

Once the extinction training trials were complete, we tested the capability of A to represent the outcome features. Following successful acquisi-

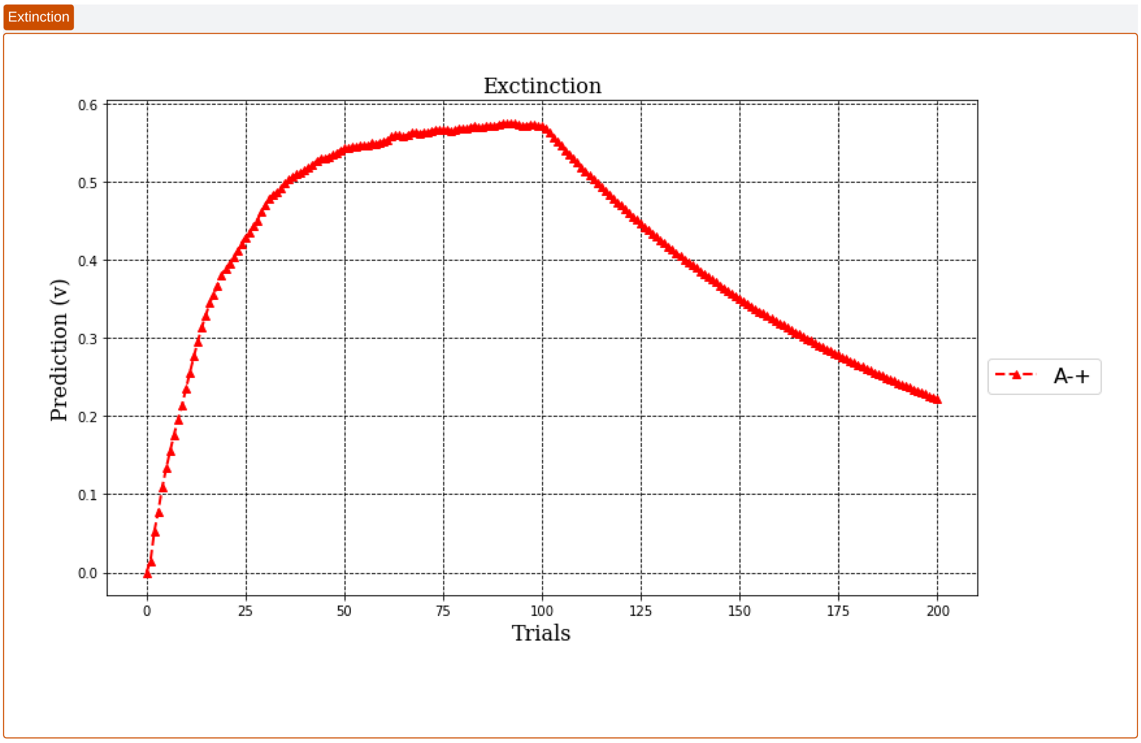


Figure 5.7: Extinction learning curves showing two distinct phases of learning. During Phase 1 (trials 1-100), standard acquisition occurs with the characteristic negatively accelerated curve reaching $\boldsymbol{V \sim 0.58}$. The phase transition at trial 100 marks the beginning of extinction. During Phase 2 (trials 101-200), CS-alone presentations cause gradual decrease in associative strength to $\boldsymbol{V \sim 0.22}$. Note the asymmetric nature of learning where extinction proceeds more slowly than acquisition and remains incomplete after 100 trials, consistent with empirical findings in the extinction literature.

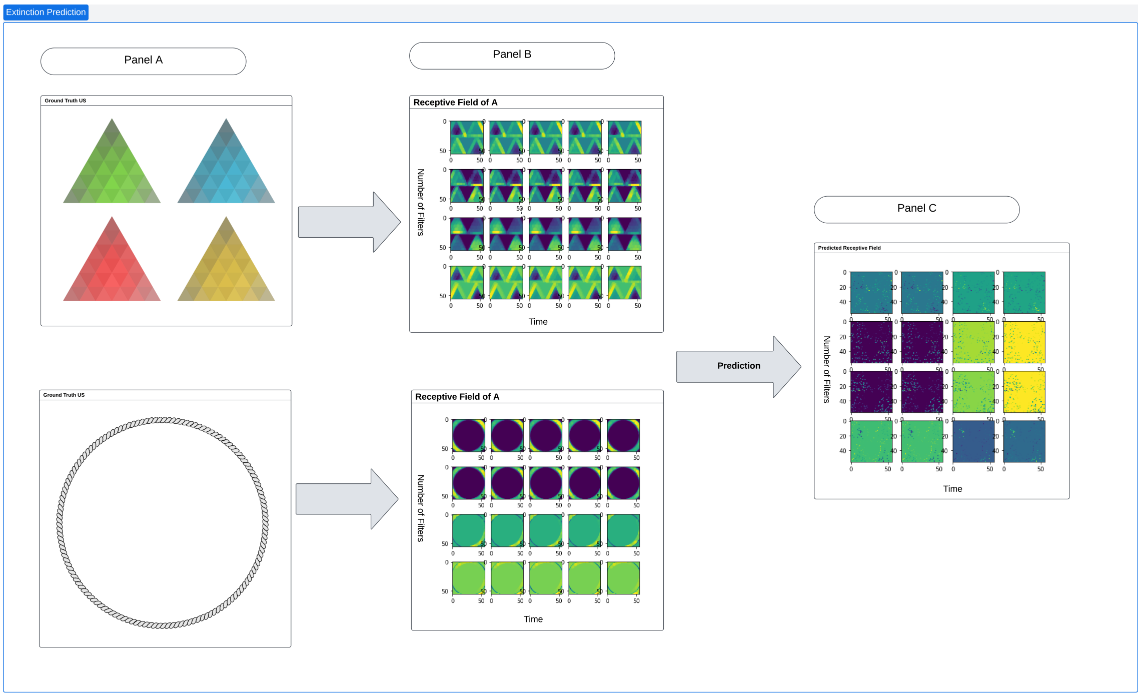


Figure 5.8: Receptive field changes during extinction. Panel A shows original stimuli. Panel B displays receptive fields before conditioning, showing the initial CNN-extracted features. Panel C reveals the critical extinction effect: after extinction, fields show degradation of both outcome features and original CS features, with increased noise and loss of coherent patterns. This suggests extinction involves active modification rather than simple decay of associations.

tion, we know from our learning curves that A had incorporated outcome features, reaching $V \sim 0.58$. However, after 100 extinction trials, significant changes occur in the receptive field structure shown in Panel C.

The primary observation is not simply the loss of circular outcome features, but a general degradation of representational clarity. The fields exhibit increased noise, reduced activation intensity and loss of coherent spatial patterns. The triangular CS features become blurred and less distinct, while residual circular patterns can still be faintly detected in some fields. This suggests extinction creates a distinct representational state rather than simply reversing acquisition.

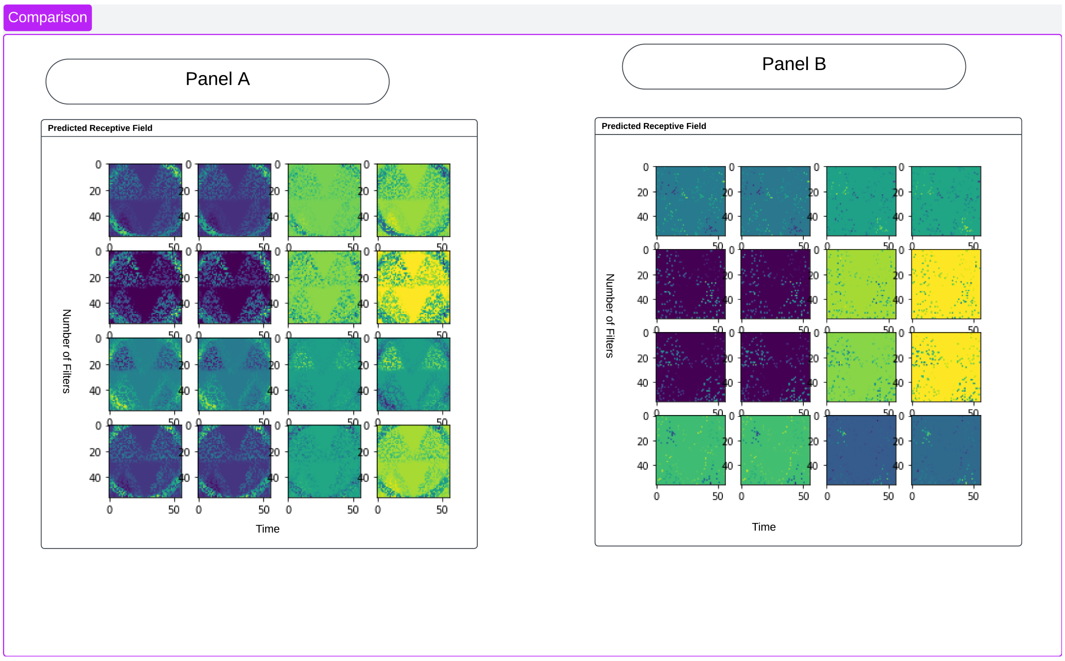


Figure 5.9: Direct comparison of receptive fields after acquisition (Panel A) versus after extinction (Panel B). Panel A shows clear, well-defined patterns with integrated circular features across multiple time points. Panel B reveals the profound impact of extinction: emergence of noisy, less coherent activation patterns. The contrast demonstrates that extinction modifies representations at a fundamental level. Note the shift from bright, focused activation (yellow-green) to diffuse, weak activation (dark purple-blue).

The comparison between receptive fields after acquisition and after extinction provides crucial insights into the extinction process. The extinction fields show increased variability across time points and feature detectors, suggesting that extinction destabilizes the learned representation rather than cleanly removing outcome features. Despite 100 extinction trials, the receptive fields do not return to their pre-conditioning state. This aligns with the incomplete extinction observed in the V values (0.22 rather than 0).

The degradation appears most severe in early time points, with some structure remaining in later processing stages. This temporal pattern may reflect different mechanisms operating at different stages of stimulus processing. These findings support theories suggesting extinction involves new learning rather than unlearning, creating modifications at a fundamental representational level. This hypothesis is supported by the asymmetric extinction rate as measured by the cue-outcome associative strength, which shows resistance to complete extinction.

5.4.4 Discussion

The extinction results reveal two key findings. First, the incomplete reduction in associative strength ($\boldsymbol{V = 0.22}$ rather than 0) aligns with empirical evidence that extinction involves new inhibitory learning rather than unlearning. Second, the degraded receptive fields with increased noise suggest that extinction creates an ambiguous representational state rather than reverting to the original stimulus representation. This supports inhibitory learning theories and explains why extinguished responses often spontaneously recover.

5.5 Blocking

5.5.1 Experimental Design

The following experimental design tested one of the most paradigmatic associative learning phenomena: Blocking. The presence of blocking is considered critical in assessing the associative nature of a phenomenon [168]. When an association between a CS and $outcome$ is sufficiently established, the acquisition predictive value by a new predictor paired with the original and the same outcome is prevented or 'blocked'. This effect is often compared to the acquisition of a compound formed by two novel predictions.

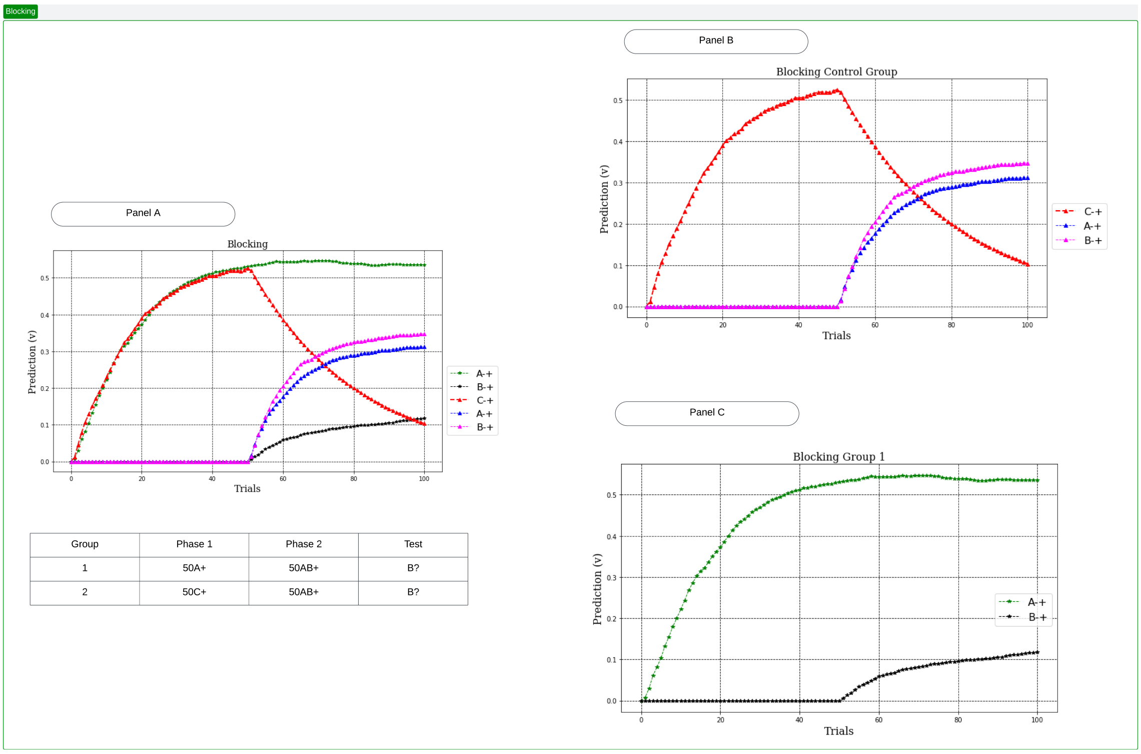
The images used in this experiment have a base of (256*256) and a receptive field of (4,56,56). We set up an experiment containing two groups, each with two phases Table 5.4.

Group	**Phase 1**	**Phase 2**	**Test**
1	50A+	50AB+	B?
2	50C+	50AB+	B?

Table 5.4: Blocking Design. Group 1 tests blocking where prior learning about A prevents learning about B. Group 2 serves as control where B can acquire associative strength normally.

In Group 1, Phase 1, a CS A - an image display containing four triangles - was paired with a $outcome$, a circle image. In Phase 2, stimulus A was accompanied by another stimulus B, forming a triangle-cat compound and paired with the same $outcome$. Each of these phases consisted of fifty training trials or epochs. On Phase 3, a single test trial of B was presented. Group 2, a control condition, was identical to Group 1 except for Phase 1, in which a novel C stimulus, a flat landscape, was presented instead of cue A.

5.5.2 Results and Evaluation



Group	Phase 1	Phase 2	Test
1	50A+	50AB+	B?
2	50C+	50AB+	B?

Figure 5.10: Blocking results demonstrating learning between experimental and control groups. Panel A shows the overlaid comparison where the difference between blocked B image (black line, $\boldsymbol{V} \sim \mathbf{0.11}$) and control B (pink line, $\boldsymbol{V} \sim \mathbf{0.36}$) is evidenced by a reduction in the strength of learning. Note the mediated extinction of C (red line declining) during Phase 2, demonstrating the model's sensitivity to associative associations. Panel B displays the control Group 2 where B acquires substantial associative strength when A is not pre-trained. Panel C shows the blocking effect in Group 1: A (green line) acquires strong associative strength in Phase 1, then when AB is presented in Phase 2, B (black line) shows minimal learning, confirming the blocking phenomenon.

The results obtained in this simulation are shown in Figure 5.10. Panel A shows all conditions overlaid for comparison, while Panels B and C display the individual group results. During the first phase of conditioning (trials

1-50), stimulus A in Group 1 and stimulus C in Group 2 both acquired positive predictive values.

The critical blocking effect emerges in Phase 2. In Group 1 (Panel C), where A had already been established as a predictor of the outcome, the added stimulus B acquired minimal predictive value (black line plateauing at $\boldsymbol{V \sim 0.11}$). This contrasts with Group 2 (Panel B), where B reached $\boldsymbol{V \sim 0.36}$ (pink line), representing a reduction in learning due to blocking.

Please note the decay in the predictive value of the control stimulus C (red line in Panel A). The model predicts contextual activation of C during Phase 2, resulting in its value being reduced through mediated learning.

5.5.3 Receptive Field Analysis

Once the training trials were completed, we tested the ability of B to predict the visual features of the outcome by extracting the predicted receptive field in both groups and comparing them.

Figure 5.11 shows the receptive field maps in Group 1. Panel B displays the CNN-extracted features before conditioning, while Panel C shows the predicted receptive field for stimulus B after the blocking phase. The predicted field shows predominantly dark purple activation, indicating minimal feature detection. The absence of circular patterns demonstrates that B failed to form predictive associations with the outcome despite being presented with it 50 times in compound with A. Some faint cat-like fea-

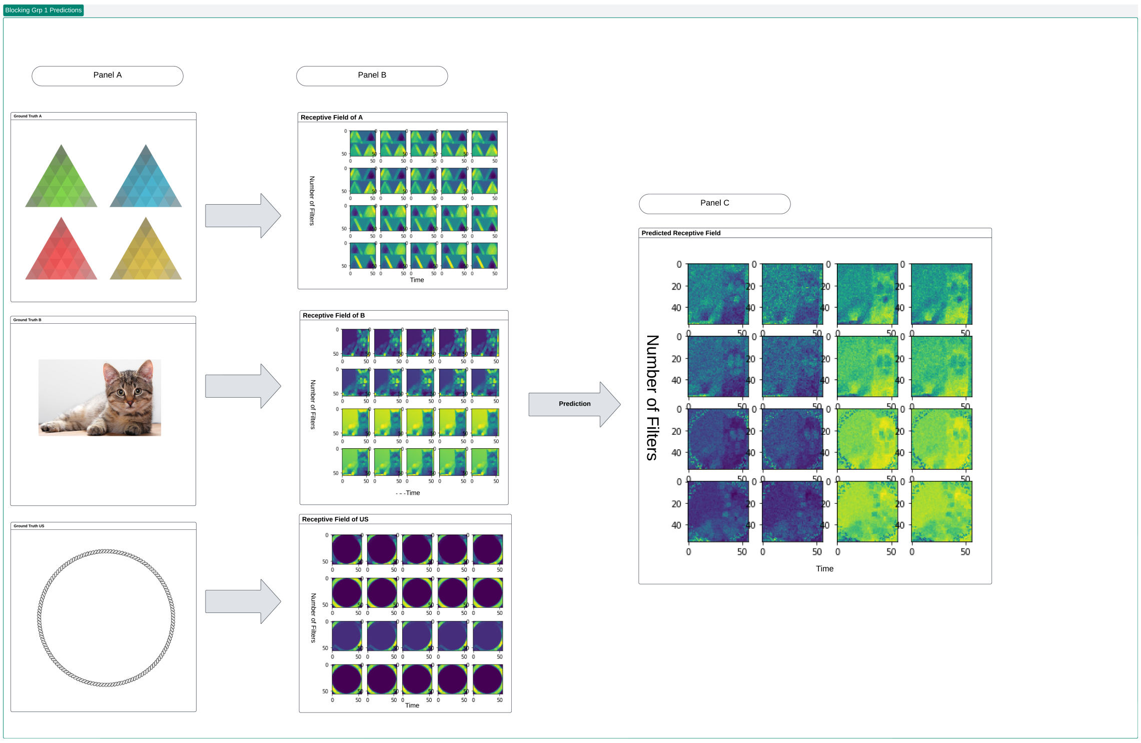


Figure 5.11: Blocking Group (Group 1) receptive field analysis. Panel A displays the original stimuli: triangles (A), cat (B), and circle outcome. Panel B shows CNN-extracted features before conditioning. Panel C reveals the critical blocking effect in the predicted receptive field of B: despite 50 AB+ trials, B's representation shows minimal circular outcome features. The receptive field remains dominated by dark purple activation with only faint traces of the cat's features visible. Notably absent are the bright yellow-green circular patterns that would indicate successful outcome prediction. This visual evidence confirms that prior learning about A prevented B from forming an effective predictive representation.

tures can be detected, suggesting the model learned B's identity ($B \to B$ associations) but not its predictive relationship with the outcome.

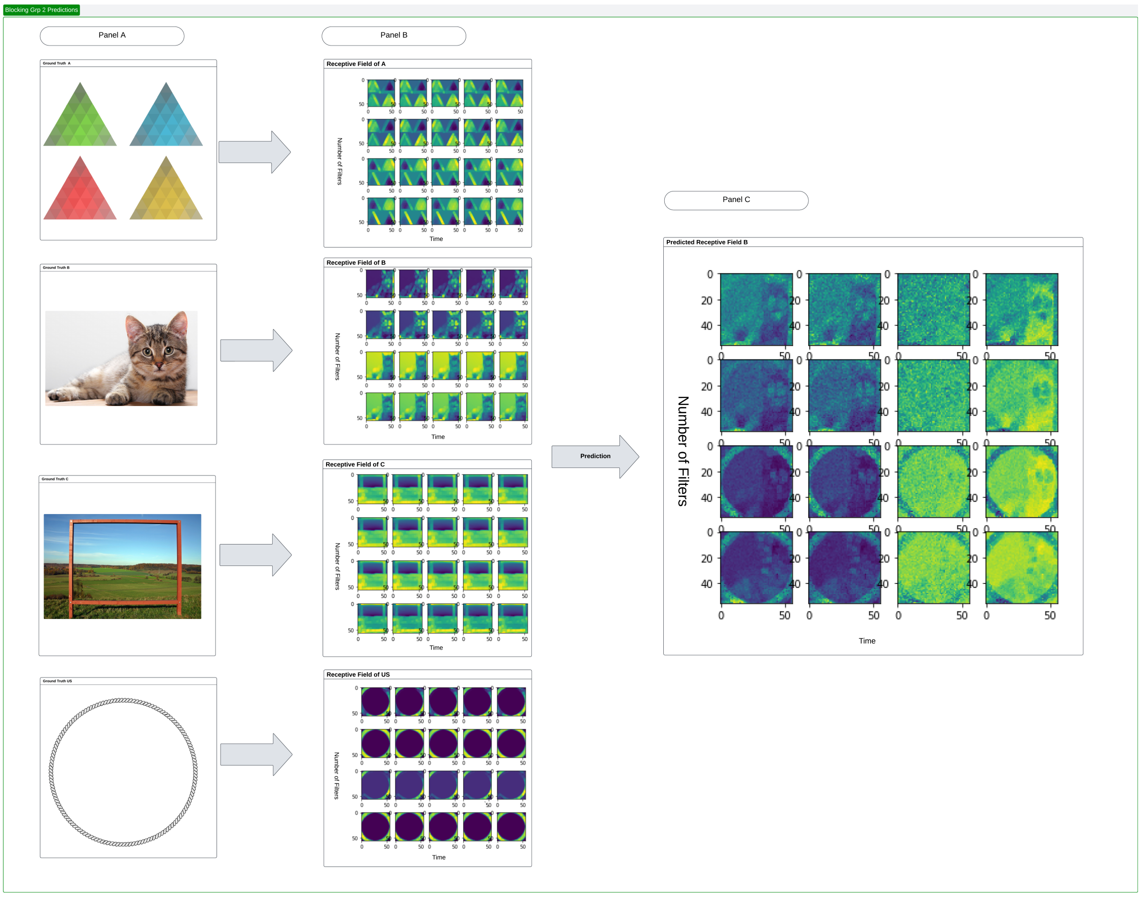


Figure 5.12: Control Group (Group 2) receptive field analysis demonstrating successful learning. Panel A shows all stimuli including the control landscape stimulus (C). Panel B displays pre-conditioning features. Panel C presents B's predicted receptive field after AB+ training, revealing differences from the blocking group. Clear circular patterns (bright yellow-green) are evident across multiple time points, particularly in later columns. The successful integration of outcome features confirms that B formed strong predictive associations when not blocked by prior learning. The contrast between Groups 1 and 2 provides strong visual evidence for the blocking phenomenon at the representational level.

Figure 5.12 shows the receptive field maps in Group 2. The left panel shows the receptive field maps at the beginning of the conditioning trials for each stimulus. The receptive field map generated by stimulus B during the test phase presents a different picture from Group 1. In the control condition, B's receptive field shows robust learning with clear evidence of outcome feature integration. In contrast to Group 1, B's receptive field in the control condition shows robust learning with clear circular patterns (bright yellow-green) visible across the receptive field. These circular features match those seen in successful acquisition, confirming that B has learned to predict the outcome when not blocked by prior learning.

The contrast between Groups 1 and 2 provides visual confirmation of the blocking effect. Group 1's B shows minimal activation and no circular features, while Group 2's B displays bright, coherent circular patterns. This representational difference corresponds to the reduction in associative strength and demonstrates that blocking operates at the level of perceptual representation formation.

These results emphasise the associative nature of the representation learning algorithms of the model, replicating a blocking phenomenon where well-established predictors of an outcome succeed in preventing a novel cue from acquiring predictive capabilities of the same outcome, hence hindering the representation of the outcome features in the formation of a compound representation.

5.5.4 Discussion

The reduction in learning for the blocked stimulus B demonstrates robust cue competition within the empirically observed range. The receptive field analysis provides novel insights into the blocking mechanism: B fails to incorporate outcome features despite the $50AB+$ trials. This visual evidence supports the theory that blocking prevents perceptual learning. The preservation of B's identity features (cat) while lacking outcome features (circle) suggests that blocking specifically disrupts predictive associations while maintaining stimulus identification. This finding could not be revealed by traditional models that only report on associative strengths.

5.6 Conditioned Inhibition

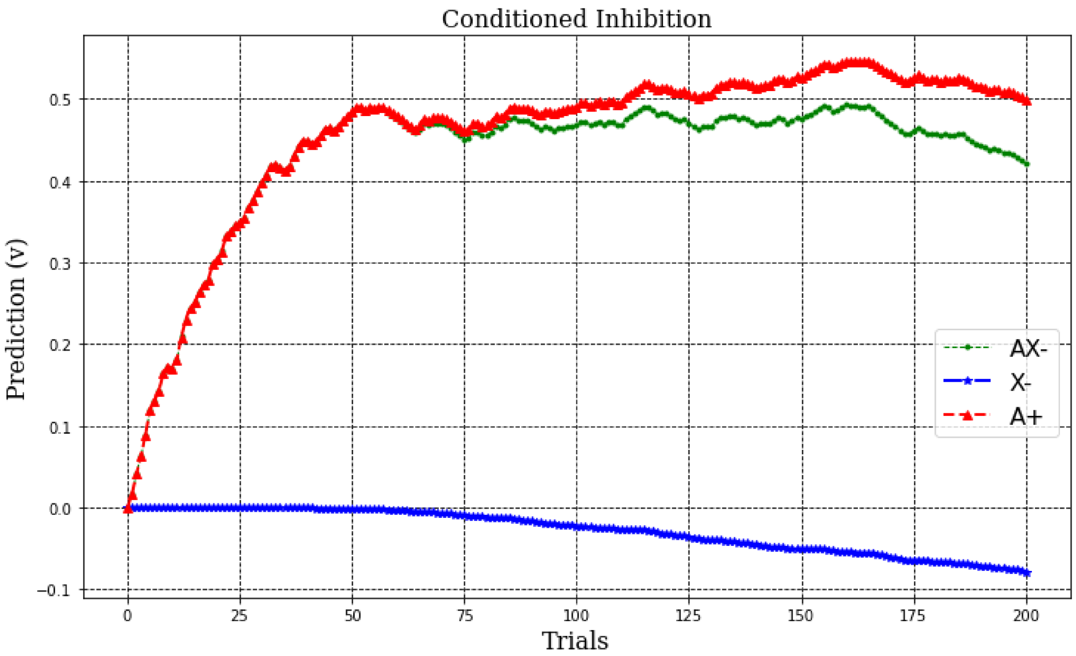
5.6.1 Experimental Design

Conditioned inhibition refers to a Pavlovian phenomenon in which the prediction of a stimulus of the outcome is hindered by the presence of another stimulus which is assumed to inhibit the representation of the outcome. The experimental design we used consisted of a single phase of training in which two types of stimuli, A+ and AX-, were presented as a series of random epochs (Table 5.5). A single test trial of AX- followed training to capture the corresponding receptive field map. The images used in this experiment have a base size of (256*256) and a receptive field size of (4, 56, 56). The setup of the experiment was a single group with a single phase.

Group	**Phase 1**	**Test**
1	200A+/200AX-	AX?

Table 5.5: Conditioned Inhibition Design. Intermixed trials of A+ and AX- train X as an inhibitor.

Conditioned Inhibition



Group 1	Phase 1	Test
1	100A+/100Ax-	AX?

Figure 5.13: Conditioned inhibition learning curves demonstrating the development of inhibitory associations. Stimulus A (red line) maintains positive associative strength $(\boldsymbol{V} \sim \mathbf{0.55})$ throughout training, showing typical acquisition for reinforced trials. X (blue line) develops negative associative strength, reaching $\boldsymbol{V} \sim -\mathbf{0.08}$, confirming the model's ability to produce inhibitory learning. The compound AX (green line) maintains intermediate strength $(\boldsymbol{V} \sim \mathbf{0.45})$. While X's inhibitory effect is modest, it demonstrates that the model can develop negative associative values without hard-coding.

5.6.2 Results and Evaluation

The two types of random trials were as follows: one consisted of $\boldsymbol{A}$, an image display containing four triangles, being paired with an outcome, a circle image Figure 5.14, and the second type of trials consisted of a compound stimulus, $\boldsymbol{AX-}$, which includes a triangle and a natural landscape being paired with no $\boldsymbol{US}$. Each of these trials runs for 200 epochs.

The learning dynamics reveal important aspects of inhibitory learning. During the initial trials (0-50), all stimuli show exploratory changes as the model determines the contingencies. A shows rapid acquisition typical of excitatory conditioning. Most critically, X progressively develops negative associative strength, crossing into negative values around trial 75 and continuing to decrease throughout training. This genuine negative V value distinguishes conditioned inhibition from simple extinction or reduced excitation.

During learning, $\boldsymbol{A+}$ acquires associative strength while $\boldsymbol{AX-}$ decreases in strength. Since the model abides by a summation principle, this entails that $\boldsymbol{V}$ of the stimulus $\boldsymbol{X}$ acquires a negative value, effectively becoming an inhibitor of the strength of another predictor of the same outcome. Figure 5.13 shows this pattern of changes in the associative strength of the stimuli. Stimulus A acquired a strong predictive value of the outcome (red line). The presence of X (green line) in compound with A reduced the prediction value of the latter. Finally, a simulation of the prediction carried by X (blue line) shows how this progressively becomes negative. Under these parameters, the condition inhibition effect is small but unequivocal.

The performance of the model was evidence that the system can produce both negative and positive $\boldsymbol{V}$ predictive values, that is, inhibition, without the need to hard-code it. In accordance with the associative summation assumption, the total prediction of the compound stimuli is the sum of the compounded CSs. The values of the compound $\boldsymbol{AX}-$ change during the trials, where the $\boldsymbol{V}$ value in $\boldsymbol{X}$ and the $\boldsymbol{V}$ value in $\boldsymbol{A}$, when combined, give us the $\boldsymbol{V}$ value of the compound $\boldsymbol{AX}$.

5.6.3 Receptive Field Analysis

Once the training trials were complete, we tested the performance of the model on the compound $\boldsymbol{AX}$. Panel B shows the receptive fields before conditioning trials, representing the initial CNN-extracted features. Panel C shows the predicted receptive field of the compound $\boldsymbol{AX}$.

The compound representation shows reduced circular patterns compared to what would be expected from A alone. Both triangular patterns (from A) and landscape features (from X) remain visible in the compound representation, while outcome features are suppressed. The receptive fields show moderate activation levels (green-blue colors) rather than the bright yellow-green of successful excitatory conditioning. This pattern is maintained across all time points.

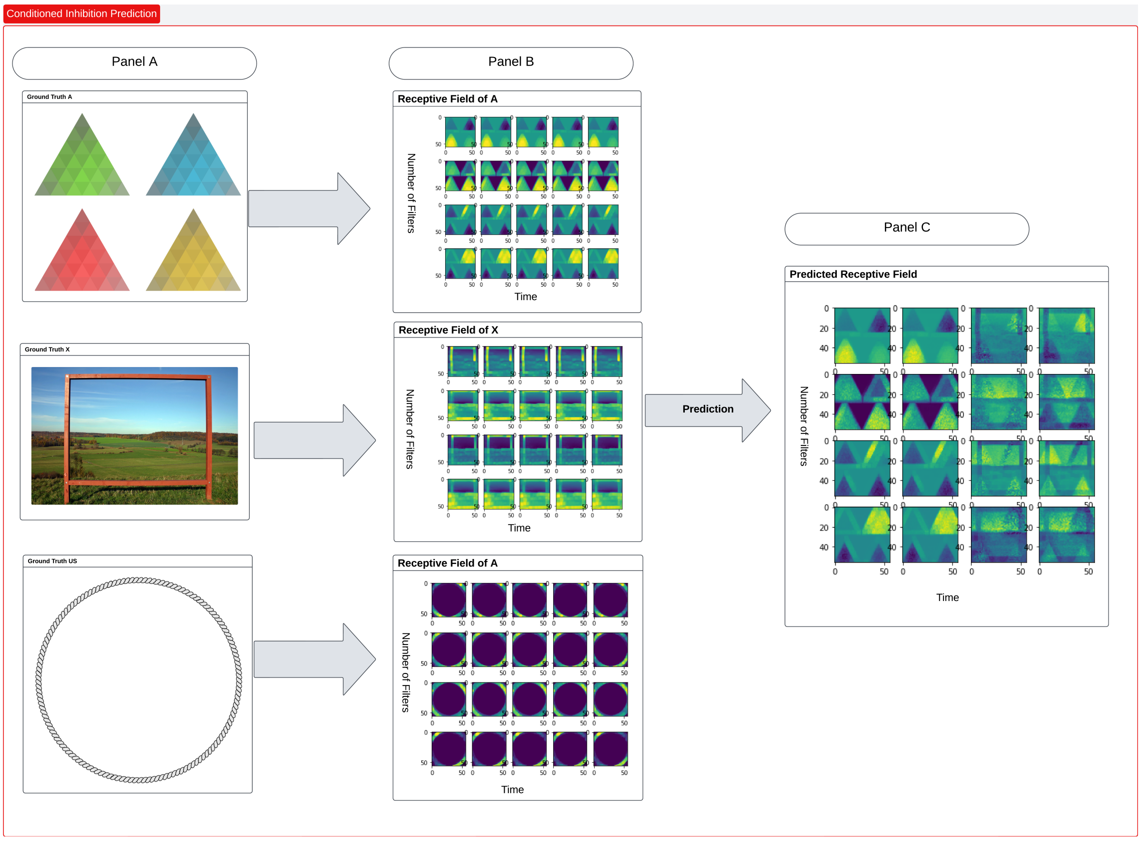


Figure 5.14: Conditioned inhibition receptive field analysis. Panel A shows the original stimuli: triangles (A), landscape (X), and circle outcome. Panel B displays preconditioning receptive fields with clear feature detection for each stimulus. Panel C presents the test result: the predicted receptive field for compound AX shows reduced outcome features. While A predicts the outcome $(\boldsymbol{V} \sim \mathbf{0.55})$, circular patterns are suppressed in the compound representation. The fields show triangular and landscape features with moderate activation (green-blue regions), suggesting that X prevents full outcome representation.

The representation shows clear features of both predictors A and X, but reduced outcome features. This evidence suggests that X has developed inhibitory properties, though the effect is modest given X's relatively small negative value $V \sim -0.08$. The visual evidence confirms that our model can implement conditioned inhibition through suppression of outcome representations at the perceptual level.

5.6.4 Discussion

Although X developed only modest negative associative strength $(V = -0.08)$, the receptive field analysis confirms genuine inhibitory learning through active suppression of outcome features in the AX compound. The relatively weak inhibition compared to theoretical models may reflect our biologically-inspired architecture, where negative values emerge from the learning dynamics rather than being hard-coded. The suppression visible in the compound representation validates that X functions as a conditioned inhibitor, preventing outcome representation even in the presence of the excitatory stimulus A. This shows that our model is capable of producing inhibitory phenomena without hardcoding inhibitory units.

5.7 Negative Patterning

5.7.1 Experimental Design

Having demonstrated that the model is capable of reproducing classic associative phenomena while building complex stimulus representation, it is now necessary to test whether the model is capable of representing non-linear relationships.

Real-world relationships are often more complex than the simple linear relationship posited by elemental models of conditioning and their summation assumption. On many occasions, these are more appropriately characterised as nonlinear conditional probabilities. Hence, learning models must incorporate processes to approximate these conditional relations. Several have been proposed to account for nonlinear discrimination learning, all of which rely on postulating *ex nihilo* representation of the stimuli.

A benchmark of nonlinear discriminative tasks is negative patterning (NP) [20, 21]. In this procedure, individual presentations of two cues, A and B, are followed by an outcome, whereas compound presentations of the same cues are not, AB. The difficulty and importance of it lie in learning that compounding the cues does not convey additive prediction, requiring a breakdown of linearity on the compound trials.

The experimental design we used to run the negative patterning trials tested the phenomenon of how the model learns to respond to a presented compound stimulus ($\boldsymbol{AB}$) and how it responds when the elements are presented separately. The trials, as organized in Table 5.6, were presented as a series of randomized epochs. The images used in this experiment have a base size of (256*256) and a receptive field size of (4, 56, 56). The setup of the experiment was a single group with a single phase; each trial in the phase was presented separately but in a random sequence.

Group	Phase 1
1	200A+/200B+/200AB-

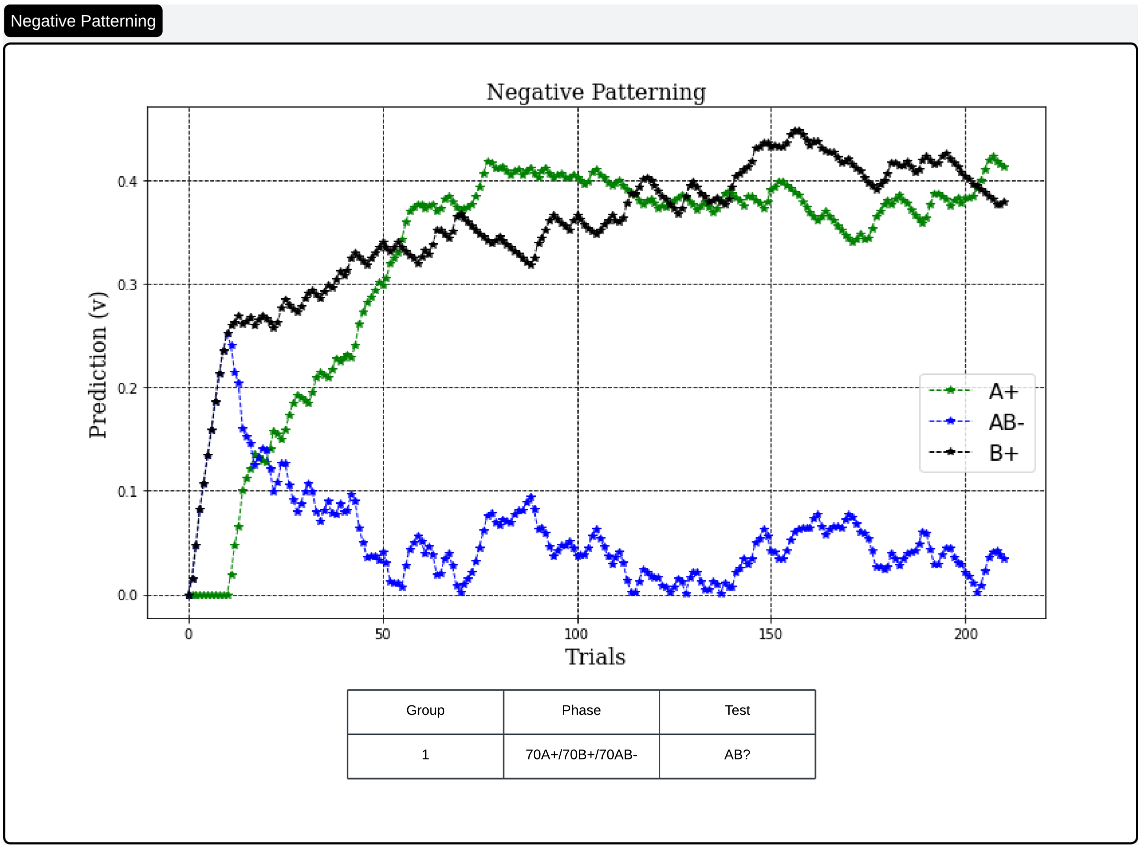
Table 5.6: Negative Patterning Design. Elements are reinforced but their compound is not.

5.7.2 Results and Evaluation

There were three random trials in this experiment: one consisted of $\boldsymbol{CS}$ A, an image display containing four triangles, paired with a $\boldsymbol{US}$, a circle image Figure 5.16; the second consisted of $\boldsymbol{CS}$ B, an image display containing a cat, paired with a $\boldsymbol{US}$, a circle image Figure 5.16; and the third random trial consisted of a compound stimulus, $\boldsymbol{AB-}$, which was a combination of triangle and cat images, paired with no $\boldsymbol{US}$. Each of these trials ran separately for 200 epochs.

The learning dynamics reveal important characteristics of how the model solves this non-linear problem. During early trials (0-50), all stimuli show exploratory fluctuations as the model samples the contingencies. The compound **AB** initially increases before declining in the first 20 trials, suggesting the model first treats it as the sum of its elements before learning the $\boldsymbol{AB-}$ trials. By trial 100, clear differentiation emerges with elements maintaining positive values while the compound shows minimal associative strength.

During conditioning, both $\boldsymbol{A+}$ and $\boldsymbol{B+}$ acquired associative strength, while $\boldsymbol{AB-}$ decreased in strength after the first few epochs. Here, the model has learned to respond to $\boldsymbol{A}$ and $\boldsymbol{B}$ separately, as well as to $\boldsymbol{AB-}$. The plots shown in **??** provide evidence that the model can discriminate



Group	Phase	Test
1	70A+/70B+/70AB-	AB?

Figure 5.15: Negative patterning results demonstrating successful non-linear discrimination. The learning curves reveal a complex pattern that cannot be explained by simple elemental summation. Both A (green line) and B (black line) acquire positive associative strength $(\boldsymbol{V} \sim \mathbf{0.41})$. Critically, the compound AB (blue line) shows different learning dynamics: after initial exploration, it maintains near-zero values around $\boldsymbol{V} \sim \mathbf{0.05}$. The separation between element and compound values demonstrates the model's ability to treat AB as a unique configuration rather than the sum of its parts. If the model operated purely on elemental summation, AB should have $\boldsymbol{V} \sim \mathbf{0.82}$ (sum of A and B); instead, it maintains near-zero associative strength, confirming successful negative patterning discrimination.

between separate associations and compound associations. If this were not the case, $AB-$ would sum to produce more responding to AB than to the individual $A+$ and $B+$. This phenomenon confirms that our model can represent configural cues.

5.7.3 Receptive Field Analysis

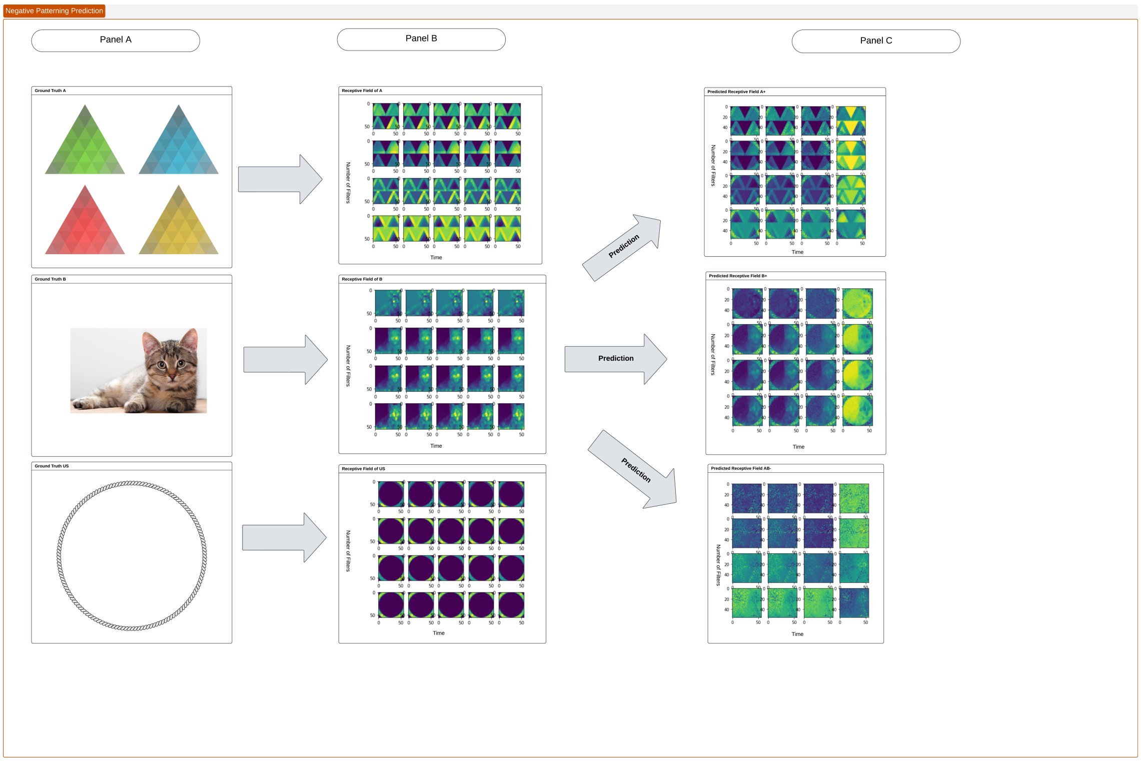


Figure 5.16: Negative patterning receptive field analysis revealing emergent configural representation. Panel A shows the original stimuli used in training. Panel B displays the pre-conditioning receptive fields extracted by the CNN, showing clear feature detection for triangles (A), cat (B), and circle outcome. Panel C presents the critical results: predicted receptive fields for A+, B+, and AB- after training. The individual elements (A+ and B+) maintain recognizable features with circular outcome patterns visible. In contrast, the compound AB- shows a degraded representation with predominantly dark purple/blue activation and no clear outcome features. This shows how the model solves negative patterning: AB develops a degraded representation that differs from both A and B individually.

Once the training trials were complete, we tested the performance of the model on all stimuli. Panel B shows the receptive fields before conditioning trials are run. These are internal representations of the visual stimuli after the CNN layer computation. Panel C shows the predicted receptive fields after training for A+, B+, and the compound AB-.

The receptive field analysis provides crucial insights into how the model solves negative patterning. The most striking feature is the differential representation between elements and compound. Where A and B individually maintain features including circular outcome patterns, the compound AB shows only diffuse, weak activation with no coherent pattern structure.

If the model were simply summing elemental representations, we would expect to see both triangular and cat features in the compound, possibly with circular outcome features. Instead, the AB representation appears to suppress all recognizable features, creating a novel representational state dominated by dark purple and blue regions.

It can be seen that when both A and B are presented together, the compound $\boldsymbol{AB}$ develops a degraded representation distinct from either element. This demonstrates how our elemental model can solve non-linear discriminations through emergent configural properties. By developing a unique representation for AB that differs qualitatively from both its elements and their sum, the model treats the compound as a distinct stimulus.

5.7.4 Discussion

The successful negative patterning discrimination provides strong evidence for emergent configural processing in our elemental model. The degraded representation of AB lacking features from either element or outcome, shows that the compound develops its own unique representation rather than summing its components. This solves the non-linear discrimination without requiring pre-specified configural units. This finding supports theories suggesting that configural processing can emerge from elemental architectures through interactive mechanisms, validating our approach of integrating CNN feature extraction with associative learning.

Chapter 6

Discussion and Future work

6.1 Discussion

In this work, we have demonstrated that integrating CNN-based visual processing with associative learning mechanisms successfully creates a unified model capable of forming complex stimulus representations. The model, based on the DDA framework [10], achieves the main objective of this thesis: simulating how visual stimuli form complex representations through associative mechanisms.

The model uses actual visual stimuli processed through CNN layers, contrasting with traditional associative models that rely on abstract symbolic representations. This approach allows the model to work with realistic perceptual inputs while maintaining the computational tractability of elemental models. The CNN layers extract hierarchical features that serve as elements for the associative mechanism, naturally providing both common and unique features across stimuli.

Our results across five fundamental phenomena, acquisition, extinction, blocking, conditioned inhibition, and negative patterning, demonstrate that the model reproduces established patterns of associative learning. The V values obtained match predictions from classical theories, while the extracted receptive fields provide direct visualization of how representation formation changes through learning.

Critically, the model provides evidence supporting the proposal by Mondragón et al. [9] that integrating CNNs with associative mechanisms enables elemental models to solve non-linear discriminations. The negative patterning results demonstrate this capability: despite using purely elemental mechanisms, the model successfully discriminates between elements and their compound. This occurs through the CNN's extraction of unique and common features, which the associative mechanism then strengthens or weakens based on their predictive value, creating distinct representational clusters.

The ability to extract and visualize learned representations sets this model apart from existing approaches. Traditional models report only abstract associative strengths, while our model shows how perceptual representations themselves are modified through learning. This provides insights into theoretical questions about how associative mechanisms shape perception.

6.1.1 Future Work

The model provides a foundation for investigating fundamental associative learning mechanisms and extracting complex stimulus representations. Several directions warrant further investigation.

First, the model would benefit from quantitative methods to evaluate predicted receptive fields. While associative strength values provide behavioral measures of learning, the visual representations extracted by our model require systematic evaluation. Information-theoretic metrics such as entropy and mutual information could quantify representational changes during learning. Additionally, diffusion models [169] could reconstruct the extracted features into high-quality images for detailed pattern analysis.

Second, the model could be extended to investigate additional associative phenomena not explored in this work. Phenomena such as overshadowing, superconditioning, and second-order conditioning would test the generality of our approach. More complex discriminations, including biconditional tasks and occasion setting, would further evaluate the model's capacity for configural representation.

Finally, the model offers a unique opportunity to examine how perceptual and associative processes interact during representation formation. By manipulating visual similarity between stimuli, we could test predictions about generalization gradients and discrimination learning. This approach could provide insights into how the cognitive system constructs representations of the environment through experience [30].

These extensions would strengthen our understanding of how associative mechanisms operate on realistic perceptual inputs, bridging the gap between abstract learning theories and biological vision systems.

Chapter A

Mathematical Derivations

A.1 Convolutional Layer Computations

A.1.1 Convolution Operation Definition

The convolution operation is implemented as a function that is defined as:

$$\text{Conv}(\text{Input}, \text{Kernel}) = \text{Output} \tag{A.1}$$

The image is resized to an appropriate shape, $(\boldsymbol{H} \times \boldsymbol{W})$, and the channel is set to 3 because we are going to work mostly with coloured images. The kernel has been set as a probability density function of weights that have exactly the same number of channels as the image.

A.1.2 Output Dimension Calculation

To obtain the output, we begin with a computation of the output shape that establishes the dimension of the placeholder for the output values that we expect from the convolution operation. The shape of the feature map is given as:

$$O = \left\{ \frac{I + 2p - K}{s} + 1 \right\} \tag{A.2}$$

where O is the output shape, I is the input shape, p is padding, K is kernel shape and s is the stride.

An example would be:

$$O = \begin{cases} \frac{56+2\times0-3}{1} + 1 \\ \frac{53}{1} + 1 \\ 54 \end{cases} \tag{A.3}$$

This output represents a weighted average image that has the shape (f_n, f_h, f_w). The number of convolution operations performed is based on the number of kernels set, which gives the number of feature maps we will have at the end of the layer.

A.1.3 Element-wise Multiplication

The implemented convolution operation is represented by:

$$a_{ik}b_{il} = \begin{cases} a_{0k} \times b_{0l} + a_{1k} \times b_{1l} \\ \sum_i a_{ik}b_{il} \\ a^T[k] \cdot b^T[l] \end{cases} \tag{A.4}$$

where i is the index of the matrix and k and l are the free indices.

A.2 Pooling Layer Computation

The pooling layer in this architecture has been implemented with the aim of reducing the dimensionality $(\boldsymbol{H} \times \boldsymbol{W})$ of the image, but not the number of channels. The operation applied is average pooling, with different strides applied on the different layers of the model.

The output dimension is first computed to set up a placeholder for the output as follows:

$$\boldsymbol{O} = \frac{\boldsymbol{I} - \boldsymbol{K}}{\boldsymbol{s}} + \boldsymbol{1} \tag{A.5}$$

Where $\boldsymbol{O}$ is the output shape, $\boldsymbol{I}$ is the input shape, $\boldsymbol{K}$ is the kernel shape and $\boldsymbol{s}$ is the stride. The operation is executed for the entire image resulting in a reduced feature map that contains the defined maximum values of the kernel operation.

A.3 Activation Function Details

A.3.1 Temporal Width Calculation

The trial duration set for each element at time $\boldsymbol{t}$ can be represented as:

$$\boldsymbol{e}^{t}_{pvo} = (\sqrt{\boldsymbol{e}^{t}} \times \boldsymbol{\delta})^{2} \tag{A.6}$$

where $\boldsymbol{e}_{t}$ is the element index, and $\boldsymbol{\delta}$ is the standard deviation of the element. Concurrently, at the same time point $\boldsymbol{t}$, the difference in time $\boldsymbol{t}$

and e^t is squared and multiplied by a skew factor k, allowing the curve to have a positive skew-shaped curve while the intensity I adjusts the amount of activity range.

A.3.2 Complete Activation Function Derivation

The gradual growth or decay of the activity curve of an element is found by calculating the negative exponent, which prevents the program from encountering a zero division error. The final activation function of a single element at t is written as:

$$A^t_{pvo} = \exp\left(-\frac{(t - e^t)^2 k}{e^t_{pvo}}\right) \times I \tag{A.7}$$

The advantage of having this function is that each element can be set with a different skew factor that controls the activation tail and a different standard deviation, δ, allowing the stimulus to generally activate with some degree of variability in its elements.

A.3.3 Stochastic Sub-element Activation

During the period of experience, the activation of the sub-elements operates following a random stochastic process that involves randomly selecting elements e^t_i from the clusters using random probabilities. Selection of sub-elements is assigned a maximum activation value of 1; if not, the values remain at 0. Eventually, all the sub-elements are assigned a value of 0 after the offset time point t_{off} of that element e_i (see Algorithm B.1 in Appendix B for detailed implementation).

A.4 Dynamic Asymptote Computation

A.4.1 Lookup Table Generation

The lookup table below is generated for all activity values ranging from [0, 1]. We created this table as a visual aid to provide a glimpse of its appearance. However, computation involves many more complexities, such as activity values that have more than one floating point. We have applied the necessary adjustments to the visual aid, but accurate references are made in the computation that does not appear in the figure.

A.4.2 Full Computation

The computation of λ involves initializing the model with an upper bound value, the maximum activation level of 1 in all of our experiments, $A_{\max}$. We also need weights, $(1-\nu)$, ν, that will be applied to the overall activity of both the predictor and the outcome elements e_i^t, e_j^t:

$$\delta = \left(A_{\max} - \frac{(OA_p^t - OA_o^t)^2}{2} \right) \times \Big((\nu \times OA_o^t) + ((1-\nu) \times OA_p^t) \Big) \tag{A.8}$$

The dynamic asymptote, δ, defines an inverse relationship between two-element activities, and it is the maximal level of learning for that time step. In error correction learning, it is used as the teacher in the context of supervised learning. Therefore, the weights are adjusted according to the difference between λ and the total predicted value of the output element, multiplied by other modulating factors.

A high weight factor, $\boldsymbol{\nu}$, on the output activity value implies that learning will converge quickly because of the high error value, even if the predictor activity has low values, and vice versa. We have implemented this factor instead of having a static $\boldsymbol{\lambda}$, which does not provide an accurate maximal value for that time.

We compute two asymptotes of learning $\boldsymbol{\delta_i^t}$, and $\boldsymbol{\delta_j^t}$, one for the predicted output layer and the other for the learned CS value.

A.5 Learning Algorithm Details

A.5.1 Weight Update Rule

Learning involves an operation that changes the weight component $\{\boldsymbol{W_{p,o}^{cs}}\}$. The change of weights, an error correction learning operation, is driven by the difference between the dynamic asymptote $\boldsymbol{\delta}$ and the sum value of all the predicted elements present at that point in time multiplied by the binary value of the presence of a cluster. This difference is known as an error.

$$\boldsymbol{W_{i,j}^{t+1}} = \boldsymbol{\Delta W_{i,j}^{t}} + \boldsymbol{W_{i,j}^{t}} \tag{A.9}$$

where:

$$\boldsymbol{\Delta W^t_{i,j} = (\delta^t_o - \sum_{i\to\forall p} W_{i,j}\times e^t_i)\times(\delta^t_p - \sum_{j\to\forall o} W_{i,j}\times e^t_j)\times S_j\times S_i\times(b \iff t_{FM^t_i} >} \tag{A.10}$$

The values $\boldsymbol{S_j}$, $\boldsymbol{S_i}$ are the saliences of the stimulus, which had been initialized as stimulus parameters at the onset of the network.

The backward discount value, $\boldsymbol{b}$ only applies if the occurrence of a cluster of the predictor is greater than the cluster of the outcome in time $\boldsymbol{t_{FM^t_i} > t_{FM^t_j}}$.

A.5.2 Prediction Computation

The forward pass loops through every connecting link $\boldsymbol{W_{i,j}}$, from all the predictor elements available in the input layers to the output layer. The results are a sum of all the prediction values, $\boldsymbol{V^t_{j\to o}}$, of each element in the output layer ($\boldsymbol{o}$) at time $\boldsymbol{t}$. The prediction value is the activity of each predicting element $\boldsymbol{OA^t_p}$ multiplied by the respective weight value $\boldsymbol{W_{i,j}}$, where $\boldsymbol{i}$, $\boldsymbol{j}$ are the predicting and output elements, respectively.

$$\boldsymbol{V^t_{j\to o} = \sum_{i\to\forall p} W_{i,j} \times OA^t_p} \tag{A.11}$$

Once the predicted values have been determined, the sum weight is accumulated at the end of each time step and averaged at the end of the stimulus duration by the total number of duration time points experienced by the stimulus. This ensures that there is a cumulative moving average of weights from one trial to another.

Chapter B

Algorithmic Procedures

B.1 Random Probability Generation

Algorithm 1 Generate Random Probabilities for samples of (FM)

```
Require: $FM.size = (m \times n), 0 \leq ab \leq 1$
Ensure: $f(x) = \frac{1}{(b-a)}$                                ▷ uniform distribution
  $elem \leftarrow dict$
  $Y \leftarrow (m, n)$
  for coordinates in $Y$ do
      $P_I \leftarrow$ Random.Sample($f(x)$)                 ▷ Inactive state $0 \leq ab \leq 1$
      $P_A \leftarrow$ Random.Sample($f(x)$)                 ▷ Active State $0 \leq ab \leq 1$
      $C \leftarrow [P_1, P_2]$
      $elem$[coordinates] $\leftarrow (C)$
  end for
```

B.2 Stimulus Activation

Algorithm 2 Stimulus Activation (FM)

```
Require: $FM.size = (m \times n), t_{\text{onset}}, t_{\text{offset}}$
Ensure: $A^t_{pvo}$                                          ▷ Element activation function
  $act_{cont} \leftarrow dict$                               ▷ activation value $[0 - 1]$
  $marker_{cont} \leftarrow dict$                            ▷ mark of selection $(0|1)$
  for ∀ time steps ∈ range($t_{\text{offset}} - t_{\text{onset}}$) do
      $e_{cont} \leftarrow e^{\text{time steps}}$                     ▷ Assign list of elements
  end for
  for ∀ time steps ∈ range($t_{\text{offset}} - t_{\text{onset}}$) do
      for ∀ ele in $e_{cont}$ do
          $act_{cont} \leftarrow A^t_{pvo}$          ▷ Apply activation function list of elements
      end for
  end for
```

B.3 Dense Layer Activation

Algorithm 3 Dense Layer Activation ($\boldsymbol{FM}$)

```
Require: $FM.size = (m \times n), t_{\text{onset}}, t_{\text{offset}}$
Ensure: $e^{t}_{pvo}$        ▷ initialize unit function for all elements at all time points
  $act \leftarrow list$        ▷ activation value $[0 - 1]$
  for ∀ time steps ∈ range($t_{\text{offset}} - t_{\text{onset}}$) do
      for ∀ element units in $e^{t}_{pvo}$ do
          $act \leftarrow$ activate()        ▷ append the activity
      end for
  end for
```

Chapter C

Implementation Details

C.1 CNN Architecture Specification

Table C.1: Complete CNN Layer Configuration

Layer	Type	Filters	Kernel	Stride	Padding	Output
0	Input	-	-	-	-	256×256×3
1	Conv2D	4	3×3	1	valid	254×254×4
2	Conv2D	4	3×3	1	valid	252×252×4
3	MaxPool	-	4×4	2	-	125×125×4
4	Conv2D	4	3×3	1	valid	123×123×4
5	Conv2D	4	3×3	1	valid	121×121×4
6	Conv2D	4	3×3	1	valid	119×119×4
7	MaxPool	-	4×4	2	-	58×58×4
8	MaxPool	-	4×4	1	-	56×56×4
9	MaxPool	-	4×4	1	-	56×56×4

C.2 Model Parameters

Table C.2: Default Parameter Values

Parameter	Symbol	Default	Description
Learning rate	α	0.1	Weight update rate
Salience	S	0.5	Stimulus salience
Temporal spread	δ	1.0	Activation temporal width
Skew factor	k	1.0	Temporal asymmetry
Intensity	I	1.0	Maximum activation
Asymptote weight	ν	0.5	Predictor/outcome balance
Backward discount	b	0.8	Temporal discount
Max activation	$A_{\max}$	1.0	Activation ceiling

C.3 Network Object Structure

The network object is an engine that applies all the operations designed to learn experiments. The configuration of the network that will run a given trial involves several components:

C.3.1 Group Object

The group object is designed to perform dynamic tasks that enable the model to perform several operations based on the specified tests and experiments. The operations of the group object are encapsulated so that the public interface available to the user requires four main inputs:

- Name of the group
- Number of Phases
- Model object
- **kwargs Stimulus configurations

The name of the group is stored in string format, a fundamental requirement of the network operations, as this is added to an overall dictionary of groups that the model will eventually run. It also takes the number of phases and stores this as an integer, which becomes useful when building results data arrays.

C.3.2 Trial Object

A trial is implemented as a placeholder object that stores cues in a set. Suppose we have a trial $\mathbf{3}\boldsymbol{A}+$, where A is a conditioned stimulus (CS) and

$+$ is an unconditioned stimulus (US). The set in this trial will store all the cues in the global cues database and ensure there is no duplication of trials.

C.3.3 Phase Object

The phase object is central to running simulations. Its main functionality involves setting up the simulation's parameters and trial sequences. It also hosts the main algorithm that runs the simulation process, making this object highly dependent on other objects, especially the computational elements applied in the computation loop.

Bibliography

[1] A. Messinger, L. R. Squire, S. M. Zola, and T. D. Albright, “Neuronal representations of stimulus associations develop in the temporal lobe during learning,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 98, no. 21, pp. 12 239–12 244, Oct. 2001.

[2] D. R. Shanks, D. Charles, R. J. Darby, and A. Azmi, “Configural processes in human associative learning,” *J. Exp. Psychol. Learn. Mem. Cogn.*, vol. 24, no. 6, pp. 1353–1378, Nov. 1998.

[3] L. Veit, G. Pidpruzhnykova, and A. Nieder, “Associative learning rapidly establishes neuronal representations of upcoming behavioral choices in crows,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 112, no. 49, pp. 15 208–15 213, Dec. 2015.

[4] W. K. Estes, “Toward a statistical theory of learning,” pp. 94–107, 1950.

[5] M. A. Arbib and J. J. Bonaiuto, *From neuron to cognition via computational neuroscience*. MIT Press, 2016.

[6] E. H. Vogel, M. E. Castro, and M. A. Saavedra, “Quantitative models of pavlovian conditioning,” *Brain Res. Bull.*, vol. 63, no. 3, pp. 173–202, Apr. 2004.

[7] E. H. Vogel, F. P. Ponce, and A. R. Wagner, “The development and present status of the SOP model of associative learning,” *Q. J. Exp. Psychol.*, vol. 72, no. 2, pp. 346–374, Feb. 2019.

[8] G. Hall, “Perceptual and associative learning,” *Oxford psychology series*, 1991.

[9] E. Mondragón, E. Alonso, and N. Kokkola, “Associative learning should go deep,” *Trends Cogn. Sci.*, vol. 21, no. 11, pp. 822–825, Nov. 2017.

[10] N. H. Kokkola, E. Mondragón, and E. Alonso, “A double error dynamic asymptote model of associative learning,” *Psychol. Rev.*, vol. 126, no. 4, pp. 506–549, Jul. 2019.

[11] R. A. Rescorla, A. R. Wagner, and Others, “A theory of pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement,” *Classical conditioning II: Current research and theory*, vol. 2, pp. 64–99, 1972.

[12] A. R. Wagner and R. A. Rescorla, “Inhibition in pavlovian conditioning: Application of a theory,” *Inhibition and learning*, pp. 301–336, 1972.

[13] J. M. Pearce, “A model for stimulus generalization in pavlovian conditioning,” *Psychol. Rev.*, vol. 94, no. 1, pp. 61–73, Jan. 1987.

[14] ——, “Similarity and discrimination: a selective review and a connectionist model,” *Psychol. Rev.*, vol. 101, no. 4, pp. 587–607, Oct. 1994.

[15] D. O. Hebb, “John wiley & sons; 1949,” *The organization of behavior: A neuropsychological approach*, 1949.

[16] D. A. Medler, “A brief history of connectionism,” *Neural Computing Surveys*, vol. 1, pp. 18–72, 1998.

[17] A. R. Rescorla, “A theory of pavlovian conditioning : Variations in the effectiveness of reinforcement and nonreinforcement,” *Current research and theory*, pp. 64–99, 1972.

[18] J. L. Kamin, “Selective association and conditioning,” *Fundamental issues in associative learning*, pp. 42–64, 1969.

[19] E. J. Kehoe and I. Gormezano, “Configuration and combination laws in conditioning with compound stimuli,” *Psychol. Bull.*, vol. 87, no. 2, pp. 351–378, 1980.

[20] J. A. Harris, “Elemental representations of stimuli in associative learning,” *Psychol. Rev.*, vol. 113, no. 3, pp. 584–605, Jul. 2006.

[21] J. A. Harris, E. J. Livesey, S. Gharaei, and R. F. Westbrook, “Negative patterning is easier than a biconditional discrimination,” *J. Exp. Psychol. Anim. Behav. Process.*, vol. 34, no. 4, pp. 494–500, Oct. 2008.

[22] S. I. Gallant, *Neural network learning and expert systems*, ser. MIT Press. Cambridge, MA: Bradford Books, Jun. 2019.

[23] K. Aizawa, “Connectionism and artificial intelligence: history and philosophical interpretation,” *J. Exp. Theor. Artif. Intell.*, vol. 4, no. 4, pp. 295–313, Oct. 1992.

[24] M. C. Mozer, P. W. Halligan, and J. C. Marshall, “The end of the

line for a brain-damaged model of unilateral neglect," *J. Cogn. Neurosci.*, vol. 9, no. 2, pp. 171–190, Mar. 1997.

[25] J. W. Garson, "Terence horgan and john tienson, connectionism and the philosophy of psychology. cambridge, MA: MIT press, 1996, cloth £27.95," *Br. J. Philos. Sci.*, vol. 50, no. 2, pp. 319–323, Jun. 1999.

[26] White, "Economic prediction using neural networks: the case of IBM daily stock returns," in *IEEE International Conference on Neural Networks*. IEEE, 1988.

[27] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 79, no. 8, pp. 2554–2558, Apr. 1982.

[28] D. Pickles, W. Bechtel, and A. Abrahamson, "Connectionism and the mind: An introduction to parallel processing in networks," *Philos. Q.*, vol. 42, no. 166, p. 101, Jan. 1992.

[29] D. E. Rumelhart, G. E. Hinton, J. L. McClelland, and Others, "A general framework for parallel distributed processing," *Parallel distributed processing: Explorations in the microstructure of cognition*, vol. 1, no. 45-76, p. 26, 1986.

[30] R. A. Rescorla, "Pavlovian conditioning. it's not what you think it is," *Am. Psychol.*, vol. 43, no. 3, pp. 151–160, Mar. 1988.

[31] ——, "Behavioral studies of pavlovian conditioning," *Annu. Rev. Neurosci.*, vol. 11, pp. 329–352, 1988.

[32] R. S. Sutton, A. G. Barto, Co-Director Autonomous Learning Laboratory Andrew G Barto, and F. Bach, *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[33] R. S. Sutton and A. G. Barto, "Toward a modern theory of adaptive networks: expectation and prediction," *Psychol. Rev.*, vol. 88, no. 2, pp. 135–170, Mar. 1981.

[34] P. C. Holland, "Event representation in pavlovian conditioning: image and action," *Cognition*, vol. 37, no. 1-2, pp. 105–131, Nov. 1990.

[35] E. C. Tolman and E. Brunswik, "The organism and the causal texture of the environment," *Psychol. Rev.*, vol. 42, no. 1, p. 43, 1935.

[36] E. L. Thorndike, "Animal intelligence; experimental studies, by edward l. thorndike," 1911.

[37] C. L. Hull, "Principles of behavior: an introduction to behavior theory," vol. 422, 1943.

[38] B. F. Skinner, *The Behaviour of Organisms*, 1938.

[39] E. C. Tolman, *Purposive Behavior in Animals and Men*. University of California Press.

[40] R. R. Bush and F. Mosteller, "Stochastic models for learning," 1955.

[41] R. C. Atkinson and W. K. Estes, "Stimulus sampling theory," 1962.

[42] H. Gulliksen, "A rational equation of the learning curve based on thorndike's law of effect," *J. Gen. Psychol.*, vol. 11, no. 2, pp. 395–434, Oct. 1934.

[43] H. Gulliksen and D. L. Wolfle, “A theory of learning and transfer: I,” *Psychometrika*, 1938.

[44] I. P. Pavlov, *Conditioned reflexes*. Oxford University Press, 1927.

[45] M. D. Egger and N. E. Miller, “Secondary reinforcement in rats as a function of information value and reliability of the stimulus,” *J. Exp. Psychol.*, vol. 64, pp. 97–104, Aug. 1962.

[46] L. J. Kamin, “Attention-like processes in classical conditioning,” 1967.

[47] ——, “Predictability, surprise, attention, and conditioning,” 1967.

[48] N. J. Mackintosh, “A theory of attention: variations in the associability of stimuli with reinforcement,” *Psychol. Rev.*, vol. 82, no. 4, p. 276, 1975.

[49] R. A. Rescorla, “Probability of shock in the presence and absence of CS in fear conditioning,” *J. Comp. Physiol. Psychol.*, vol. 66, no. 1, pp. 1–5, Aug. 1968.

[50] A. R. Wagner, “Incidental stimuli and discrimination learning,” *Animal discrimination learning*, pp. 83–111, 1969.

[51] I. P. Pavlov, *Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex*. Translated and edited by Anrep, GV (Oxford University Press, London, 1927), 1927.

[52] A. R. Wagner, F. A. Logan, K. Haberlandt, and T. Price, “Stimu-

lus selection in animal discrimination learning," *J. Exp. Psychol.*, vol. 76, no. 2, pp. 171–180, Feb. 1968.

[53] B. Widrow and M. E. Hoff, "Adaptive switching circuits," 1960.

[54] W. F. Hill and R. A. Rescorla, "Pavlovian Second-Order conditioning: Studies in associative learning," p. 372, 1981.

[55] S. Reiss and A. R. Wagner, "CS habituation produces a "latent inhibition effect" but no active "conditioned inhibition"," *Learn. Motiv.*, vol. 3, no. 3, pp. 237–245, Aug. 1972.

[56] C. L. Hull, "Principles of behavior: An introduction to behavior theory, Appleton-Century-Crofts, new york, 1943," *Google Scholar*.

[57] A. Dickinson, G. Hall, and N. J. Mackintosh, "Surprise and the attenuation of blocking," *J. Exp. Psychol. Anim. Behav. Process.*, vol. 2, no. 4, pp. 313–322, Oct. 1976.

[58] C. Bonardi and G. Hall, "Learned irrelevance: No more than the sum of CS and US preexposure effects?" *J. Exp. Psychol. Anim. Behav. Process.*, vol. 22, no. 2, pp. 183–191, Apr. 1996.

[59] N. J. Mackintosh, "Stimulus selection: Learning to ignore stimuli that predict no change in reinforcement," *Constraints on learning: Limitations and predispositions.*, vol. 488, 1973.

[60] M. E. Le Pelley, "The role of associative history in models of associative learning: a selective review and a hybrid model," *Q. J. Exp. Psychol. B*, vol. 57, no. 3, pp. 193–243, Jul. 2004.

[61] J. M. Pearce and G. Hall, “A model for pavlovian learning: variations in the effectiveness of conditioned but not of unconditioned stimuli,” *Psychol. Rev.*, vol. 87, no. 6, pp. 532–552, Nov. 1980.

[62] G. Hall and J. M. Pearce, “Latent inhibition of a CS during CS–US pairings,” *J. Exp. Psychol. Anim. Behav. Process.*, vol. 5, no. 1, p. 31, 1979.

[63] J. Konorski, “Integrative activity of the brain; an interdisciplinary approach.”

[64] A. R. Wagner, “Context-sensitive elemental theory,” *The Quarterly Journal of Experimental Psychology Section B*, vol. 56, no. 1, pp. 7–29, Feb. 2003.

[65] A. R. Wagner and S. E. Brandon, “A componential theory of pavlovian conditioning. handbook of contemporary learning theories, eds mowrer RR, klien SB,” 2001.

[66] S. E. Brandon, E. H. Vogel, and A. R. Wagner, “A componential view of configural cues in generalization and discrimination in pavlovian conditioning,” *Behav. Brain Res.*, vol. 110, no. 1-2, pp. 67–72, Jun. 2000.

[67] C. L. Hull, “The discrimination of stimulus configurations and the hypothesis of afferent neural interaction,” *Psychol. Rev.*, vol. 52, no. 3, pp. 133–142, May 1945.

[68] M. E. Bitterman, “Classical conditioning in the goldfish as a func-

tion of the CS-UCS interval," *J. Comp. Physiol. Psychol.*, vol. 58, no. 3, p. 359, 1964.

[69] M. Davis, L. S. Schlesinger, and C. A. Sorenson, "Temporal specificity of fear conditioning: effects of different conditioned stimulus-unconditioned stimulus intervals on the fear-potentiated startle effect," *J. Exp. Psychol. Anim. Behav. Process.*, vol. 15, no. 4, pp. 295–310, Oct. 1989.

[70] C. R. Gallistel and J. Gibbon, "Time, rate, and conditioning," pp. 289–344, 2000.

[71] M. D. Mauk and B. P. Ruiz, "Learning-dependent timing of pavlovian eyelid responses: differential conditioning using multiple interstimulus intervals," *Behav. Neurosci.*, vol. 106, no. 4, pp. 666–681, Aug. 1992.

[72] D. R. Williams, "Classical conditioning and incentive motivation," *Classical conditioning. New York: Appleton-Century-Crofts*, vol. 1, no. 5, 1965.

[73] E. J. Kehoe, "Classical conditioning: fundamental issues for adaptive network models," *Learning and computational neuroscience: Foundations of adaptive networks*, pp. 390–420, 1990.

[74] N. Schneiderman, "Interstimulus interval function of the nictitating membrane response of the rabbit under delay versus trace conditioning," *J. Comp. Physiol. Psychol.*, vol. 62, no. 3, pp. 397–402, Dec. 1966.

[75] N. Schneiderman and I. Gormezano, “CONDITIONING OF THE NICTITATING MEMBRANE OF THE RABBIT AS a FUNCTION OF CS-US INTERVAL,” *J. Comp. Physiol. Psychol.*, vol. 57, pp. 188–195, Apr. 1964.

[76] M. C. Smith, S. R. Coleman, and I. Gormezano, “Classical conditioning of the rabbit’s nictitating membrane response at backward, simultaneous, and forward CS-US intervals,” pp. 226–231, 1969.

[77] N. A. Schmajuk, *Animal Learning and Cognition: A Neural Network Approach.* Cambridge University Press, Apr. 1997.

[78] S. E. Brandon, E. H. Vogel, and A. R. Wagner, “Computational theories of classical conditioning,” in *A Neuroscientist’s Guide to Classical Conditioning*, J. W. Moore, Ed. New York, NY: Springer New York, 2002, pp. 232–310.

[79] R. S. Sutton and A. G. Barto, “Time-derivative models of pavlovian reinforcement,” in *Learning and computational neuroscience: Foundations of adaptive networks , (pp*, M. Gabriel, Ed. Cambridge, MA, US: The MIT Press, xv, 1990, vol. 613, pp. 497–537.

[80] A. Harry Klopf, “A neuronal model of classical conditioning,” *Psychobiology*, vol. 16, no. 2, pp. 85–125, Jun. 1988.

[81] A. H. Klopf, “7 - classical conditioning phenomena predicted by a Drive-Reinforcement model of neuronal function,” in *Neural Models of Plasticity*, J. H. Byrne and W. O. Berry, Eds. Academic Press, Jan. 1989, pp. 104–132.

[82] J. W. Moore, J. E. Desmond, N. E. Berthier, D. E. Blazis, R. S. Sutton, and A. G. Barto, “Simulation of the classically conditioned nictitating membrane response by a neuron-like adaptive element: response topography, neuronal firing, and interstimulus intervals,” *Behav. Brain Res.*, vol. 21, no. 2, pp. 143–154, Aug. 1986.

[83] A. R. Wagner, “SOP: a model of automatic memory processing in animal behavior.(in) information processing in animals: memory mechanisms.(ed.) NE spear, RR miller,” 1981.

[84] J. E. Mazur and A. R. Wagner, “An episodic model of associative learning,” *Quantitative analyses of behavior: Acquisition*, vol. 3, pp. 3–39, 1982.

[85] E. Mondragón, J. Gray, E. Alonso, C. Bonardi, and D. J. Jennings, “SSCC TD: a serial and simultaneous configural-cue compound stimuli representation for temporal difference learning,” *PLoS One*, vol. 9, no. 7, p. e102469, Jul. 2014.

[86] R. A. Rescorla, ““ configural” conditioning in discrete-trial bar pressing,” *J. Comp. Physiol. Psychol.*, vol. 79, no. 2, p. 307, 1972.

[87] N. A. Schmajuk and J. J. DiCarlo, “Stimulus configuration, classical conditioning, and hippocampal function,” *Psychol. Rev.*, vol. 99, no. 2, pp. 268–305, Apr. 1992.

[88] J. A. Lamoureux, C. V. Buhusi, and N. A. Schmajuk, “A real-time theory of pavlovian conditioning: Simple stimuli and occasion setters,” in *Occasion setting: Associative learning and cognition in an-*

imals , (pp, N. A. Schmajuk, Ed. Washington, DC, US: American Psychological Association, xxi, 1998, vol. 440, pp. 383–424.

[89] N. A. Schmajuk, J. A. Lamoureux, and P. C. Holland, "Occasion setting: a neural network approach," *Psychol. Rev.*, vol. 105, no. 1, pp. 3–32, Jan. 1998.

[90] R. A. Rescorla, "Conditioned inhibition and facilitation," *Information processing in animals: Conditioned inhibition*, pp. 299–326, 1985.

[91] R. T. Ross and P. C. Holland, "Conditioning of simultaneous and serial feature-positive discriminations," *Anim. Learn. Behav.*, vol. 9, no. 3, pp. 293–303, Sep. 1981.

[92] P. C. Holland, "Differential effects of reinforcement of an inhibitory feature after serial and simultaneous feature negative discrimination training," *J. Exp. Psychol. Anim. Behav. Process.*, vol. 10, no. 4, pp. 461–475, Oct. 1984.

[93] S. Grossberg, "A neural model of attention, reinforcement and discrimination learning," pp. 263–327, 1975.

[94] J. E. Desmond, "Temporally adaptive responses in neural models: The stimulus trace," in *Learning and computational neuroscience: Foundations of adaptive networks , (pp*, M. Gabriel, Ed. Cambridge, MA, US: The MIT Press, xv, 1990, vol. 613, pp. 421–456.

[95] J. E. Desmond and J. W. Moore, "Adaptive timing in neural net-

works: the conditioned response," *Biol. Cybern.*, vol. 58, no. 6, pp. 405–415, 1988.

[96] J. W. Moore, J. E. Desmond, and N. E. Berthier, "Adaptively timed conditioned responses and the cerebellum: a neural network approach," *Biol. Cybern.*, vol. 62, no. 1, pp. 17–28, 1989.

[97] A. R. Wagner, S. E. Brandon, S. B. Klein, and R. R. Mowrer, "Evolution of a structured connectionist model of pavlovian conditioning (AESOP)," *Contemporary learning theories: Pavlovian conditioning and the status of traditional learning theory*, pp. 149–189, 1989.

[98] N. Schneiderman, "Response system divergencies in aversive classical conditioning," *Classical conditioning II: Current research and theory*, pp. 341–376, 1972.

[99] S. L. Betts, S. E. Brandon, and A. R. Wagner, "Dissociation of the blocking of conditioned eyeblink and conditioned fear following a shift in US locus," *Anim. Learn. Behav.*, vol. 24, no. 4, pp. 459–470, Dec. 1996.

[100] S. E. Brandon, J. C. Bombace, W. A. Falls, and A. R. Wagner, "Modulation of unconditioned defensive reflexes by a putative emotive pavlovian conditioned stimulus," pp. 312–322, 1991.

[101] S. E. Brandon and A. R. Wagner, "Modulation of a discrete pavlovian conditioned reflex by a putative emotive pavlovian conditioned stimulus," *J. Exp. Psychol. Anim. Behav. Process.*, vol. 17, no. 3, pp. 299–311, Jul. 1991.

[102] ——, “Occasion setting: Influences of conditioned emotional responses and configural cues,” in *Occasion setting: Associative learning and cognition in animals , (pp*, N. A. Schmajuk, Ed. Washington, DC, US: American Psychological Association, xxi, 1998, vol. 440, pp. 343–382.

[103] J. C. Gewirtz, S. E. Brandon, and A. R. Wagner, “Modulation of the acquisition of the rabbit eyeblink conditioned response by conditioned contextual stimuli,” *J. Exp. Psychol. Anim. Behav. Process.*, vol. 24, no. 1, pp. 106–117, Jan. 1998.

[104] R. F. Thompson, N. H. Donegan, G. A. Clark, D. G. Lavond, J. S. Lincoln, J. Madden, IV, L. A. Mamounas, M. D. Mauk, and D. A. McCormick, “Neuronal substrates of discrete, defensive conditioned reflexes, conditioned fear states, and their interactions in the rabbit,” *Classical conditioning., 3rd ed.*, vol. 3, pp. 371–399, 1987.

[105] S. R. Coleman and I. Gormezano, “Classical conditioning of the rabbit’s (oryctolagus cuniculus) nictitating membrane response under symmetrical CS-US interval shifts,” pp. 447–455, 1971.

[106] W. F. Prokasy and J. D. Papsdorf, “Effects of increasing the interstimulus interval during classical conditioning of the albino rabbit,” *J. Comp. Physiol. Psychol.*, vol. 60, no. 2, pp. 249–252, Oct. 1965.

[107] J.-S. Choi and J. W. Moore, “Cerebellar neuronal activity expresses the complex topography of conditioned eyeblink responses,” *Behav. Neurosci.*, vol. 117, no. 6, pp. 1211–1219, Dec. 2003.

[108] Kelso, Kelso, and T. H. Brown, “Differential conditioning of associative synaptic enhancement in hippocampal brain slices,” pp. 85–87, 1986.

[109] J. F. Medina, K. S. Garcia, W. L. Nores, N. M. Taylor, and M. D. Mauk, “Timing mechanisms in the cerebellum: testing predictions of a large-scale computer simulation,” *J. Neurosci.*, vol. 20, no. 14, pp. 5516–5525, Jul. 2000.

[110] M. A. Gluck, E. S. Reifsnider, and R. F. Thompson, “Adaptive signal processing and the cerebellum: Models of classical conditioning and VOR adaptation,” *Neuroscience and connectionist theory*, pp. 131–185, 1990.

[111] M. D. Mauk and N. H. Donegan, “A model of pavlovian eyelid conditioning based on the synaptic organization of the cerebellum,” *Learn. Mem.*, vol. 4, no. 1, pp. 130–158, May 1997.

[112] E. H. Vogel, S. E. Brandon, and A. R. Wagner, “Stimulus representation in SOP: II. an application to inhibition of delay,” *Behav. Processes*, vol. 62, no. 1-3, pp. 27–48, Apr. 2003.

[113] “Predictive timing under temporal uncertainty: The time derivative model of the conditioned response,” in *Timing of Behavior*, D. A. Rosenbaum and C. E. Collyer, Eds. The MIT Press, 1998.

[114] R. S. Sutton and A. G. Barto, “A temporal-difference model of classical conditioning,” in *Proceedings of the ninth annual conference of the cognitive science society*, 1987, pp. 355–378.

[115] S. Grossberg and D. S. Levine, "Neural dynamics of attentionally modulated pavlovian conditioning: blocking, interstimulus interval, and secondary reinforcement," p. 5015, 1987.

[116] S. Grossberg and N. A. Schmajuk, "Neural dynamics of adaptive timing and temporal discrimination during associative learning," *Neural Netw.*, vol. 2, no. 2, pp. 79–102, Jan. 1989.

[117] P. S. Churchland and T. J. Sejnowski, *The Computational Brain*. The MIT Press, 2016.

[118] Y. L. Cun, L. D. Jackel, B. Boser, J. S. Denker, H. P. Graf, I. Guyon, D. Henderson, R. E. Howard, and W. Hubbard, "Handwritten digit recognition: applications of neural network chips and automatic learning," *IEEE Commun. Mag.*, vol. 27, no. 11, pp. 41–46, Nov. 1989.

[119] J. Heaton, "Ian goodfellow, yoshua bengio, and aaron courville: Deep learning," *Genet. Program. Evolvable Mach.*, vol. 19, no. 1, pp. 305–307, Jun. 2018.

[120] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[121] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[122] B. C. Love, O. Guest, P. Slomka, V. Navarro, and E. Wasserman, "Deep networks as models of human and animal categorization," in *CogSci*, 2017.

[123] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[124] K. Simonyan and A. Zisserman, "Very deep convolutional networks for Large-Scale image recognition," Sep. 2014.

[125] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[126] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[127] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. Physiol.*, vol. 160, pp. 106–154, Jan. 1962.

[128] ——, "Ferrier lecture. functional architecture of macaque monkey visual cortex," *Proc. R. Soc. Lond. B Biol. Sci.*, vol. 198, no. 1130, pp. 1–59, Jul. 1977.

[129] Zhou and Chellappa, “Computation of optical flow using a neural network,” in *IEEE 1988 International Conference on Neural Networks*, Jul. 1988, pp. 71–78 vol.2.

[130] D. Scherer, A. Müller, and S. Behnke, “Evaluation of pooling operations in convolutional architectures for object recognition,” pp. 92–101, 2010.

[131] M. Ranzato, F. J. Huang, Y. Boureau, and Y. LeCun, “Unsupervised learning of invariant feature hierarchies with applications to object recognition,” in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2007, pp. 1–8.

[132] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, “A survey of the recent architectures of deep convolutional neural networks,” 2020.

[133] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, “End-to-end text recognition with convolutional neural networks,” in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, Nov. 2012, pp. 3304–3308.

[134] B. Xu, N. Wang, T. Chen, and M. Li, “Empirical evaluation of rectified activations in convolutional network,” May 2015.

[135] P. Ramachandran, B. Zoph, and Q. V. Le, “Searching for activation functions,” Oct. 2017.

[136] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, “Recent advances in convo-

lutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018.

[137] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Int. J. Uncertainty Fuzziness Knowledge Based Syst.*, vol. 06, no. 02, pp. 107–116, Apr. 1998.

[138] D. Misra, "Mish: A self regularized non-monotonic neural activation function," *arXiv preprint arXiv:1908.08681*, 2019.

[139] Y. LeCun, Y. Bengio, and Others, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[140] X. Zhang and Y. LeCun, "Text understanding from scratch," Feb. 2015.

[141] L. Lu, H.-C. Shin, H. R. Roth, and M. Gao, "Deep convolutional neural networks for Computer-Aided detection: CNN architectures, dataset characteristics and transfer learning deep convolutional neural networks for Computer-Aided detection: CNN architectures, dataset characteristics and transfer," *IEEE Trans. Med. Imaging*, vol. 35, no. 1285-1298, pp. 1602–03 409, 2016.

[142] M. Kafi, M. Maleki, and N. Davoodian, "Functional histology of the ovarian follicles as determined by follicular fluid concentrations of steroids and IGF-1 in camelus dromedarius," *Res. Vet. Sci.*, vol. 99, pp. 37–40, Apr. 2015.

[143] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[144] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision – ECCV 2014*. Springer International Publishing, 2014, pp. 818–833.

[145] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," Dec. 2013.

[146] F. Grün, C. Rupprecht, N. Navab, and F. Tombari, "A taxonomy and library for visualizing learned features in convolutional neural networks," Jun. 2016.

[147] Y. Bengio, "Deep learning of representations: Looking forward," in *Statistical Language and Speech Processing*. Springer Berlin Heidelberg, 2013, pp. 1–37.

[148] H. Wang and B. Raj, "On the origin of deep learning," Feb. 2017.

[149] Q. Nguyen, M. C. Mukkamala, and M. Hein, "Neural networks should be wide enough to learn disconnected decision regions," Feb. 2018.

[150] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[151] O. Delalleau and Y. Bengio, “Shallow vs. deep Sum-Product networks,” in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 666–674.

[152] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, Inception-ResNet and the impact of residual connections on learning,” Feb. 2016.

[153] C. Szegedy, V. Vanhoucke, S. Ioffe, and others, “Rethinking the inception architecture for computer vision,” *Proceedings of the*, 2016.

[154] C. Dong, C. C. Loy, K. He, and X. Tang, “Image Super-Resolution using deep convolutional networks,” pp. 295–307, 2016.

[155] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” 2017.

[156] J. Kuen, X. Kong, G. Wang, and Y.-P. Tan, “DelugeNets: Deep networks with efficient and flexible Cross-Layer information inflows,” 2017.

[157] Tong, T. Tong, G. Li, X. Liu, and Q. Gao, “Image Super-Resolution using dense skip connections,” 2017.

[158] K. Kawaguchi, J. Huang, and L. P. Kaelbling, “Effect of depth and width on local minima in deep learning,” pp. 1462–1498, 2019.

[159] S. Zagoruyko and N. Komodakis, “Wide residual networks,” 2016.

[160] D. Han, J. Kim, and J. Kim, “Deep pyramidal residual networks,” 2017.

[161] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” 2017.

[162] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, “Squeeze-and-Excitation networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020.

[163] D. G. Lowe, “Distinctive image features from Scale-Invariant keypoints,” pp. 91–110, 2004.

[164] P. Dollar, Z. Tu, P. Perona, and S. Belongie, “Integral channel features,” 2009.

[165] N. Chouhan, A. Khan, and H.-U.-R. Khan, “Network anomaly detection using channel boosted and residual learning based deep convolutional neural network,” p. 105612, 2019.

[166] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, “Residual attention network for image classification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3156–3164.

[167] A. G. Roy, N. Navab, and C. Wachinger, “Concurrent spatial and channel ‘squeeze & excitation’ in fully convolutional networks,” pp. 421–429, 2018.

[168] D. R. Shanks, “Learning: from association to cognition,” *Annu. Rev. Psychol.*, vol. 61, no. 1, pp. 273–301, 2010.

[169] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," 2020. [Online]. Available: https://arxiv.org/abs/2006.11239