The Application of Classical Conditioning to the Machine Learning of a Commonsense Knowledge of Visual Events

Timothy Andrew Furze

Submitted in accordance with the requirements for the degree of
Doctor of Philosophy

The University of Leeds
School of Computing

July 2013

The candidate confirms that the work submitted is his own, except where work which has formed part of jointly-authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

The jointly-authored publication, detailed below, is a description of a nascent version of the work presented in this thesis. The work in the publication is entirely directly attributable to the candidate. The other author, Brandon Bennett, acted in a supervisory and editorial capacity. The reference for the publication is:

Furze, Timothy A. & Bennett, Brandon (5 April 2011). "Using the Principles of Classical Conditioning to Learn Event Sequences". In Kazakov, Dimitar & Tsoulas, George (editors), *Proceedings of the Artificial Intelligence and the Simulation of Behaviour (AISB) Conference 2011*, volume Computational Models of Cognitive Development, pages 40–47. Society for the Study of Artificial Intelligence and the Simulation of Behaviour, York, U.K.

Abstract

In the field of artificial intelligence, possession of commonsense knowledge has long been considered to be a requirementto construct a machine that possesses artificial general intelligence. The conventional approach to providing this commonsense knowledge is to manually encode the required knowledge, a process that is both tedious and costly. After an analysis of classical conditioning, it was deemed that constructing a system based upon the stimulus-stimulus interpretation of classical conditioning could allow for commonsense knowledge to be learned through a machine directly and passively observing its environment. Based upon these principles, a system was constructed that uses a stream of events, that have been observed within the environment, to learn rules regarding what event is likely to follow after the observation of another event. The system makes use of a feedback loop between three sub-systems: one that associates events that occur together, a second that accumulates evidence that a given association is significant and a third that recognises the significant associations. The recognition of past associations allows for both the creation of evidence for and against the existence of a particular association, and also allows for more complex associations to be created by treating instances of strongly associated event pairs to be themselves events. Testing the abilities of the system involved simulating the three different learning environments. The results found that measures of significance based on classical conditioning generally outperformed a probability-based measure. This thesis contributes a theory of how a stimulus-stimulus interpretation classical conditioning can be used to create commonsense knowledge and an observation that a significant sub-set of classical conditioning phenomena likely exist to aid in the elimination of noise. This thesis also represents a significant departure from existing reinforcement learning systems as the system presented in this thesis does not perform any form of action selection.

Acknowledgements

I would first like to thank my supervisor, Brandon Bennett. Not only due to the great deal of formal support that was given in a patient and reliable manner, but due to the intellectual growth that he cultivated. I always enjoyed the challenges to my ideas and debates we engaged in. I also would like to thank Brandon for getting me the opportunity in the first place. While I know I wasn't the model student and this journey has certainly had its more than its fair share of ups and downs, we got there in the end.

My family, especially my mum, my dad and my sister, deserve an enormous amount of thanks. I would not have been able to do this if it wasn't for their unquestioning love and support not just in the context of my PhD, but throughout my life.

My friends have been another source of strength that has allowed me to finish this journey. If I didn't have an escape, then I would not have been able to keep up with the amount of work that is needed. There has been a great deal specific pieces of help from my friends, though I will not give names for fear of forgetting someone, Instead I will say this: If you know me and think you deserve credit for helping me in some manner, then consider yourself thanked.

I would like to thank my viva examiners, Eduardo Alonso and John Stell, and my mock viva examiner Marc de Kamps for agreeing to take the time to read this thesis and examine me on its contents.

Throughout the course of my studies, I have been in receipt of a variety of formal and informal assistance. I would like the thank Vania Dimatrova, Tony Cohn and Derek Magee for the assistance they have provided. In addition, I have attended an assortment of workshops and my thanks go to the organisers and instructors for improving my skills and knowledge. Furthermore I would like to thank the EPSRC and therefore the U.K. taxpayer for providing me with the funding to make this research possible.

Finally, if you are reading this thesis out of your own choice, then I thank you, mysterious reader of obvious good taste, for taking an interest in my work.

– Timothy A. Furze, 26$^{\text{th}}$ July 2013

Table of Contents

List of Algorithms

List of Figures

List of Tables

Chapter 1

Introduction

Throughout the twentieth century, psychology made significant progress through studying how animals learn, with its findings significantly informing the study of human psychology. Those same findings have been a rich source of inspiration for the development of artificial intelligence systems. This thesis continues this use of psychological ideas within artificial intelligence.

This thesis contributes towards the field of artificial intelligence generally and more specifically to the sub-fields of machine learning and commonsense knowledge acquisition. This was done by applying some of the ideas from psychology to a problem found within artificial intelligence. The psychological ideas that have been used come from the field of classical conditioning. The problem within artificial intelligence is known as the knowledge acquisition bottleneck.

The development of this thesis began with a number of guiding principles, or positions, on how the general problem of artificial intelligence should be approached. These positions are stated within this chapter in order for the reader to gain insight into the basis from which this thesis was built. After stating these positions, the chapter will then give an introduction to the problem this thesis addresses. In considering the problem in reference to the positions, two hypotheses arose that this thesis seeks to test – a discussion of the hypotheses comprises the next section of this chapter. Finally, the overall structure of the thesis is described.

1.1 Positions

The positions of this thesis are those assumptions about how to approach a problem that are required, either implicitly or explicitly, for progress to occur. This implies that other positions may be just as valid, and may be held for each author's own reasons, but every research project must have them in order to

conduct the research in the first place. This is because if a project did not hold any positions, then there would not be any basis on which to make decisions in the design of the project's methodology.

For the purposes of clarity, the three positions will be stated, and then each will be discussed in turn in its own subsection:

Position A: Using the results from psychology to inform the design of artificial intelligence systems has the best likelihood of long-term success in the creation of artificial general intelligence systems.

Position B: The ability to model the environment and adapt to changes within that environment is a requirement for general intelligence.

Position C: On balance, predicate logic is the framework that is most likely to be successful in representing a complete model of an agent's environment.

1.1.1 Position A

In observing artificial intelligence systems, it appears that there are six primary sources of inspiration to the problem of designing artificial intelligence systems: Introspection, computational theory, statistics, game theory, biology & neuroscience and psychology. There are upsides and downsides to using every one of these sources of inspiration, and different people's weightings of each of those upsides and downsides could lead to a different conclusion as to which source of inspiration is likely to lead to the best results.

Inspiration from introspection refers to the approach whereby a system designer designs their system based on internal observation of their own thought processes, relying on the fact that they are themselves an example of what they seek to build. Some early research in artificial intelligence appears to be of this sort, and was discussed by McCarthy & Hayes:

> *"Programs have been written to solve a class of problems that give humans intellectual difficulty. ... In the course of designing these programs intellectual mechanisms of greater or lesser generality are identified sometimes by introspection, sometimes by mathematical analysis, and sometimes by experiments with human subjects. Testing the programs sometimes leads to better understanding of the intellectual mechanisms and the identification of new ones."*
>
> *(McCarthy & Hayes, 1969, pp. 465–466)*

In one guise or another, introspection has been long associated with the study of human thought and can be traced to Socrates (Schultz & Schultz, 1996, p. 77). The approach was used in a rigorous way in the early days of the foundation of psychology as an independent discipline and had one of the discipline's founders, Wundt, as a proponent (Schultz & Schultz, 1996, p. 77). The central argument for the use of introspection is that due to the subjective

nature of conscious experience, the only person able to observe it is the person who is having the experience. In terms of the design of artificial intelligence systems, the approach also requires little background knowledge and so allows for quick initial progress, this could be a reason for its early use within artificial intelligence. Within psychology, the approach is now largely seen as lacking rigor. The best known argument against introspection is by Watson (1913), whose main argument is that due to the subjectivity of introspection, the results gained by using it cannot be reliably confirmed. Watson argued that one should only observe behaviour and not attempt to theorise on internal mental states – a school of thought that became known as behaviourism. In more recent times, introspection in psychology is deemed unreliable as it requires a-priori knowledge of one's own unconscious thought process. It has been shown by psychological experimentation that it is not possible to know this (Nisbett & Wilson, 1977).

Computational-theoretical, statistical and game-theoretical inspiration appears to have similar upsides and downsides and so shall be discussed together. Each of these inspirations holds its basis in mathematical proof. This leads to the design of systems that can be seen to be highly rational, sometimes having provably perfect rationality. Their downside, however, is that the computational resources such systems require can make them infeasible to use in all but the smallest problem instances. This issue leads to the problem of marshalling computational resources to optimise between rationality of an action and the reaction time (Russell *et al.*, 1993).

Using biology and neuroscience for inspiration does imply at least some level of in-built optimality between rationality and response time, as arguably evolutionary forces have already had to solve the issue (though whether those optimisations can apply to any computational version remains to be seen). A benefit of using neuroscience as a source of inspiration is that it is a source of ideas for systems that have not been originated by another human. This means that arguably it provides ideas that may not occur to a human approaching the problem of artificial intelligence from either an introspective or mathematical basis. The drawback of using neuroscience or biology as a source of inspiration is that the level of abstraction is too low to easily derive intelligent behaviour. To use the well-worn brain-computer analogy, the task of neuroscience is analogous to attempting to reverse-engineer a database server by analysing the function of each individual transistor. It is theoretically possible, but would take a great deal of effort. This is not to say that the work of neuroscience should not be done, it provides a highly valuable contribution, however this thesis holds that using the field as a primary source of inspiration for artificial intelligence systems will only yield advances at a very slow rate. A better role would be to indirectly influence artificial intelligence by providing a basis and plausibility checks for psychological theories.

Designing artificial intelligence systems by drawing inspiration from psychology shares the same advantages, to differing extents, as that of biology and

neuroscience inspired systems, but does not have the drawback of an abstraction level that is too low to be of direct use. There are other, weaker drawbacks though. Psychological theories on the whole are vaguer than those of other approaches. This however can be turned into a positive point as it allows an artificial intelligence system designer enough leeway to implement the system in a manner that suits the mostly serial nature of computer processing. An analogy would be that psychological theory can be used as a form of software engineering specification. Another drawback is that due to not being able to properly control all experimental variables, data from experimental psychology is inherently noisier than the physical sciences. This leads to a greater variety of psychological theories that fit the data, but contradict one another, leading to a higher likelihood of a theory utilised by the artificial intelligence system designer being incorrect. Poole, Mackworth & Goebel (1998) make an argument against taking inspiration from psychology, neuroscience or biology. The argument is one by analogy with the development of flight:

> *"First note that there are several ways to understand flying. One is to dissect known flying animals and hypothesize their common structural features as necessary fundamental characteristics of any flying agent. With this method an examination of birds, bats, and insects would suggest that flying involves the flapping of wings made of some structure covered with feathers or a membrane. . . . An alternate methodology is to try to understand the principles of flying without restricting ourselves to natural occurrences of flying. This typically involves the construction of artefacts that embody the hypothesized principles, even if they do not behave like flying animals in any way except flying. This second method has provided both useful tools, airplanes, and a better understanding of the principles underlying flying, namely aerodynamics."*
>
> *(Poole* et al.*, 1998, pp. 2–3)*

This argument ignores the fact that it was only through the use of observations of biological flight that the principles of aerodynamics were discovered. Sir George Cayley, in publishing his findings on aerodynamics, effectively founding the field, made the observation that birds do not flap once full velocity has been reached (Cayley, 1809, p. 167). Poole *et al.*'s argument is that by only using examples from nature, it is restricting the space of intelligent entities, and this hampers progress in the development of the underlying principles. With the analogy, it is quite obvious that aeroplanes and birds use different methods of flight but both use the same underlying principles. However, the argument that the principles of aerodynamics came about due to some elementary derivation is fallacious, as was described earlier. As such, Poole *et al.*'s argument that such an approach should also apply to artificial intelligence is precarious – we don't yet know enough about the principles of intelligence to use them without reference to existing intelligent beings.

Due to the weighting placed on the advantages and disadvantages of each of these sources of inspiration, this thesis uses psychology. Ultimately, all sources of inspiration have a place in contributing towards the problem of artificial intelligence, and any full solution will be a synthesis of ideas from all fields, but from the point of view of a single project, there will always be one field that is used more than the others.

1.1.2 Position B

Consider an agent that did not have any model of its environment. Without any model, the agent would not know the likely consequences of any action it takes. This would preclude it from selecting the most rational action. General intelligence is more than just rational action selection; however it would be surprising if it was not at least part of the picture.

Assuming that the need for some form of model is a requirement, there are two ways in which an agent can come into possession of a model. Firstly it could be built in to the agent, either implicitly or explicitly, provided by an external party or mechanism – such as an agent's designer or through an evolutionary process[1]. Secondly it could learn for it itself through interacting with the environment. If the environment is dynamic (i.e. the facts and rules of the environment are subject to change), then the first system would not have any mechanism to update its environment to reflect the changes. Therefore, if the environment does change, in all likelihood the rationality of the agent's actions would diminish. Assuming that there is a mechanism for learned rules to change over time, the second method would be able to adapt and continue to produce rational actions.

Note that while the above arguments talk of rational action as a measure of intelligence, it is not the only form of intelligence that this thesis recognises. It is possible to imagine a fully passive learning system, creating a model of its environment that the system itself does not act upon, but the model instead could be queried, like a database. While no metric of rational action could find such a system to be intelligent, it can be argued to possess some form of intelligence.

Even an agent that is able to act upon its environment would still need to learn knowledge in both a passive and active manner. Furze & Bennett argued:

> *"For instance, an animal can associate the sound of a rock slide with the sight of falling rocks. It can also learn to actively avoid being hit by an incoming rock. Only when both passive and active learning are together can the animal associate the sound of a rock slide with danger, without actually being caught in a rock slide. For*

[1]It should be noted that while an evolutionary processes can create systems that learn from the environment, in this context it is only referring to immutable knowledge provided through evolution.

> *another example, consider using a hairdryer to move a toy sailing ship. For a system to be able to plan such a course of action without first observing it, the system would need to have passively associated air currents with moving sailing ships and observed that the action of activating a hairdryer causes an air current."*
>
> *(Furze & Bennett, 2011, p. 46)*

Until the definition of general intelligence has been settled, this can only be an opinion, though as a position it should be less controversial than position A. However, there are approaches that can be seen to be incompatible with this position. One prominent approach is the CYC project.

The CYC project (Lenat *et al.*, 1985; Guha & Lenat, 1990) is a project in its third decade that aims to create a knowledge database that encompasses all the knowledge that is usually implicit in human discourse, due to that knowledge being common to all but the youngest of the species. Lenat (1996) argues that once the project reaches some threshold of knowledge, then combined with a program to analyse natural language, it could add all of human knowledge to its database by in effect, reading all of it. This argument for the system gaining general intelligence-level behaviour if it becomes large enough is incompatible with the position of this thesis because this database, as it is hand written, would not be able to adapt to changes in its knowledge. Therefore any change in the "sum of all human knowledge", such as widely-held theories being disproved, could not be adapted to without further intervention.

1.1.3 Position C

To explain the reasoning for this position, first a number of measures for the worth of a particular framework shall be presented, followed by a discussion of a selection of frameworks in terms of those measures. The list of frameworks discussed is exhaustive in neither the number of frameworks nor the depth to which each framework is discussed to argue that predicate[2] logic is the best with complete authority; hence this is merely a position taken by this thesis rather than any stronger sort of assertion. As with the other positions, different weightings of the advantages and disadvantages could lead to others to take a different viewpoint and the reasoning below merely reflects the position of this thesis. Note that in order to keep the length of this section from being far longer than would be in line with its relevance, the descriptions and definitions of each of the frameworks are assumed to be known by the reader.

[2]In this thesis, predicate logic primarily refers to first-order logic. However, as this thesis does not make use of the quantification elements of first-order logic, the more general term "predicate logic" has been used. In terms of position C, it is believed that quantification will ultimately be needed even though it is not used in this thesis.

There are four measures this thesis has used to determine the worth of a particular framework:

1. **Expressiveness**. This refers to what limitations (if any) the framework has in expressing a model of an environment.

2. **Computational efficiency**. Computational efficiency refers to how much memory space a model of a given fidelity expressed in the framework takes up and how quickly a model's predictions can be retrieved, added, removed or modified.

3. **Inferential capacity**. This is how easy it is for the framework to be used to extrapolate from the environmental model that has been provided explicitly.

4. **Human readability**. This measure of human readability is how easy it is for a human to understand how the model reflects its environment and how easy it is for a human to manually modify the model.

Four frameworks are discussed: Predicate logic, neural networks, automata and Markov models. Each of these frameworks is Turing-complete and so in theory is able classify any input that a human can classify, assuming the Church-Turing thesis is correct. This means that any of the frameworks are able in theory to express any model of the environment that a human can. This may not be the case in a practical sense however.

Predicate logic is arguably the most expressive in practical terms of the three frameworks, as it is able to explicitly express all parts of an environmental model, or as they are discussed in the literature, theories (with the term model referring to a structure that satisfies a theory). Theories created within predicate logic do however vary in efficiency, dependent on the theory's complexity, but for real-world cases is arguably better than either of the other methods discussed. Predicate logic allows for considerable inferential capacity through its rules of deduction. This system of knowledge representation does also allow for a trained human to be able to both read and write knowledge in this format with ease due to the high degree of modularity in how the knowledge is represented. Alonso (2002) suggested a further benefit of predicate logic, arguing that because predicate logic is able to be easily understood by humans, this allows for agents based on predicate logic to be safer than other forms of knowledge representation.

Unlike the other three formalisms, it is less widely known that some types of neural networks are Turing complete (McCulloch & Pitts, 1943; Hyötyniemi, 1996). This fact is less useful in practice however, as it requires a complex network for even simple Turing machines. As a practical framework, neural networks are able to represent concepts that involve continuums in an efficient manner – by encoding real-valued inputs as input nodes. Neural networks are however less efficient at representing binary relationships between inputs (i.e. those relationships that exist or don't exist), as the only way to represent

a relationship between a pair of inputs is by applying a high/low weighting to every possible pair of inputs to represent whether a particular pair has that property. Neural networks are also relatively poor at allowing for the knowledge to be created within their structure as concepts can only be input into the system as examples, rather than general rules. The number of needed examples could be very high for the system to correctly identify a particular concept. Once a network is trained on a concept it is able to identify other unseen examples of a concept fairly robustly, so does demonstrate a reasonable level of inference. Neural networks are poor for human readability and the ability to be manually changed, since individual neuron weights affect the classifications in non-linear ways that are hard for a human to predict.

While automata with the addition a reversible tape for memory are by definition Turing complete, the expressiveness of automata is severely curtailed in practice. This is because of the massive proliferation of states automata require to approximate any continuum or reflect any uncertainty. Automata are a very good tool for matching particular specific patterns though, and so were a viable candidate for this thesis. For concrete pattern matching, automata are also the most efficient of all the systems described, due to only needing to record one state plus any stack or tape memory with the time taken for a single input being constant. The efficiency of changing the knowledge reflected in an automaton is better than for a neural network because it can be done without needing a large set of examples of the change, but is not as efficient as predicate logic because new knowledge necessitates at least some change in the existing structure, which is not required by predicate logic, due to the modularity inherent in simply adding an extra rule to the existing theory. Due to its discrete nature, automata cannot easily represent any form of knowledge best encoded as a real-valued property. This implies that it is inefficient to create automata that recognise patterns where a wide range of input values can cause transition to the same state. Automata have a low inferential capacity because of there being no mechanism to allow for extrapolation based on existing encoded knowledge. Human readability of automata is better than for neural networks, but is far behind predicate logic.

Markov models share a number of similarities to automata in terms of representation of knowledge; such as the need to change existing structures to accommodate new knowledge, though for some of the knowledge (such as transition probabilities) the change is minimal. One main difference, due to the probabilistic transitions in states, is that there is at least some knowledge that is best encoded as a real-valued property that can be feasibly represented by the system. The inferential capacity is greatly improved upon, with knowledge such as being able to infer the likelihood of a particular pattern occurring. The human readability of such a system is again, similar to that of automata.

1.2 The Problem

By taking the positions mentioned in the previous section, it is worth considering the problems that have been identified over time as arising from taking those positions. While it may not be possible for one person or project to tackle all of the problems associated with a given position, by holding a particular stance there is also an implied belief that all the current problems associated with a stance will be solved in time. With this in mind, this thesis looks at an attempt to partially solve a single problem connected with the stated stances. The problem that this thesis is concerned with is referred to as the "knowledge acquisition bottleneck". This problem is most associated with position C – with the literature regarding the problem coming primarily from those that share the stance, though the problem can be loosely associated with position B. Position A is more a stance of methodology in relation to this thesis and the problem it addresses, so has little association with the problem.

In order to understand the knowledge acquisition bottleneck problem, first a related concept known as commonsense knowledge needs to be described. Commonsense knowledge refers to the "every-day" knowledge that every human knows and uses, such as an expectation that when a ball is thrown into the air, it will eventually come down again; or that if someone has been walking in one direction and becomes occluded by a bush, that the person will eventually re-appear at the other side of the bush. Knowing these sorts of things is crucial if an agent is to interact within a human world, either in attempting to understand the large amount of implicit details in any form of human communication or simply just to carry out a simple robotics task in a robust manner.

The study of commonsense knowledge has been historically linked strongly with those that study how that knowledge in general can be represented and reasoned with and so is traditionally encoded in the tools favoured by that community of researchers, namely predicate logic. The amount of knowledge that commonsense knowledge covers is vast, both in breadth and depth. In addition, this form of knowledge is typically highly inter-dependent, which means that a large amount of it is needed to be encoded before it can be used. Up until the last ten years or so, commonsense knowledge was always encoded manually. Due to the fact that typically large amounts of knowledge were needed, encoding this knowledge was a task that was both highly tedious and costly – due to the task needing to be done by highly trained people. This problem of encoding commonsense knowledge is known as the knowledge acquisition bottleneck.

This thesis looks at a possible approach to addressing this problem by having machines acquire the knowledge from first-hand experience, using classical conditioning as the method of machine learning. Due to the vast area that commonsense knowledge covers, this thesis looks at a niche of commonsense knowledge – that of learning to predict certain types of visual event. These

predictions, when learned, could be argued to correspond to particular pieces of knowledge, such as "when a ball is observed to move upwards, it is to be expected that it will later move downwards". A fuller description of the knowledge acquisition bottleneck and other approaches that have been used to deal with the problem can be found in chapter three.

1.3 Hypotheses

In considering the positions taken and the problem undertaken by this thesis, two inter-related hypotheses arose. These hypotheses shall be simultaneously tested by this thesis. If they are allowed to stand after testing, then they imply that the approach to dealing with the knowledge acquisition problem this thesis takes may be valid. After stating the two hypotheses, each hypothesis shall be discussed in more detail.

Hypothesis A: The phenomena of classical conditioning can be used as a mechanism-independent specification for a system that allows an agent to learn a commonsense knowledge model of its environment.

Hypothesis B: An agent using the phenomena of classical conditioning that passively observes a dynamic environment will still be able to learn a partial commonsense knowledge model of that environment.

1.3.1 Hypothesis A

Classical conditioning is a collection of related phenomena that are based on a single process whereby two stimuli, one of which is biologically relevant to the animal, when presented to an animal together, form an association in the mind of the animal – an association that strengthens over repeated presentations. One proposal about the nature of the classical conditioning phenomena is that they are the mechanism that allows an animal to learn a model of its environment (Sokolov, 1960; Schmajuk *et al.*, 1996).

Commonsense knowledge is a field of artificial intelligence that is concerned with the representation and reasoning of the knowledge of the world that appears to be self-evident to humans, such as "Trees are usually green" and "Things that go up, later come down". When taken a whole, a commonsense knowledge base effectively comprises a model of the human environment. As such, if classical conditioning is a mechanism for learning a model of an environment, then it should be able to be used to learn commonsense knowledge. By exposing a computer program that learns through classical conditioning to a stream of events, this thesis posits that the system will produce a model of how those events interrelate.

Importantly for this use of classical conditioning to build a commonsense knowledge base, all the phenomena of classical conditioning are described in purely behavioural "input-output" terms. This means that as long as the each input gives the correct output, the process that turns one into the other does

not necessarily need to be biologically plausible. This is the reasoning for why the hypothesis states that it is believed that the classical conditioning is independent of its mechanism. This allows from a more software engineering perspective, for the list of phenomena of classical conditioning to act as if it were a feature list of a software specification. In chapter two, each of these phenomena is described and analysed in terms of how it contributes to a learning agent.

1.3.2 Hypothesis B

The proposal that classical conditioning specifies the mechanism that allows an animal to learn a model of its environment can be interpreted to imply that the phenomena of classical conditioning may not just apply when one of the stimuli is biologically relevant to the animal. The association may be more general, but instead it is that case that when an association involves a biologically relevant stimulus, that the learned association can be observed externally. This interpretation is effectively that of the stimulus-stimulus interpretation of classical conditioning, which is further described in chapter two.

A direct implication for not requiring the presence of a biologically relevant stimulus for learning to occur is that an agent should be able to learn any given association within the environment, even when the agent has not yet acted upon the environment. A consequence of this is that it should be possible for an agent to passively learn an environment and still accrue, through passive association of stimuli, knowledge of that environment.

There are two qualifiers to this argument: Firstly, the agent's environment needs to be dynamic – if an environment is sufficiently constrained that the agent is the only cause of change an environment then passive observation will not produce learning, as the environment will not change. Secondly, passive observation may not necessarily allow for a complete description of an environment. If the agent is able to change the environment in ways that would not occur without the an action arising from the agent, or areas of the environment are not able to be observed without action arising from the agent, then a complete description of the environment is not possible based on passive observation alone.

Work based upon classical conditioning from within machine learning and artificial intelligence in general has been focused on attempting to maximise rewards received from the environment based upon the agent's actions within that environment. This is usually referred to as the reinforcement learning problem. In such a problem, passive learning does not occur, as passivity does not allow for rewards to occur. Therefore, to investigate this hypothesis requires the traditional reinforcement learning problem to be abandoned. Chapter three looks at reinforcement learning in the context of work related to this thesis.

The field of artificial intelligence benefits from investigating these two hypotheses. Testing both of these hypotheses necessitates the construction of

a system that firstly implements at least some of the phenomena of classical conditioning and secondly learns passively. If either or both of the hypotheses are confirmed by testing the system, then a new learning algorithm will have been developed in the process, furthering the field of artificial intelligence.

1.4 Thesis Structure

This thesis is structured as follows. In chapter two, the background of classical conditioning and its phenomena are explained. Along-side this description, a machine learning-based analysis of classical conditioning is presented. This analysis starts with a first-principles analysis of the criteria what a system would need to do to learn about its environment passively. These criteria then are used to analyse the phenomena of classical conditioning, to demonstrate a correspondence between the two.

Chapter three discusses the work related to this thesis. First the chapter discusses attempts made to model the phenomena of classical conditioning. Next the chapter looks at the nature of commonsense knowledge. This is followed by a review of an existing form of conditioning-based machine learning known as reinforcement learning. Finally the chapter briefly reviews some of the research in computer vision into event detection and tracking.

In chapter four the workings of the system are described in detail. This includes the input processing and a description in turn of each of the four modules that the system is comprised of. Also included in chapter four is a description of the mechanism of each of the ten models of classical conditioning that have been developed.

In chapter five, the methods that are used to evaluate the system are described. This includes the systems used for producing the required input data. This is followed by chapter six which presents the results of that evaluation.

The whole thesis is then rounded off by chapter seven which includes the conclusions drawn, a review of the contributions of the thesis and a discussion of potential further research avenues this thesis highlights.

Appendix A lists the logical definitions of all the basic events that are used to create the patterns of events that are recognised by the system described by this thesis.

Appendix B provides the mathematical derivations for the equations used by some of the models of classical conditioning.

Appendix C lists that Gantt charts that were used as part of the effort to generate a proxy ground truth, as described in section 5.3.

Appendix D lists the various constants that can be set within the system, along with the values that they were set to in the evaluation.

Chapter 2

Classical Conditioning and its Analysis

The ideas that this thesis contributes towards machine learning, and artificial intelligence more generally, are firmly rooted in the ideas of classical conditioning. This chapter will both introduce and analyse those ideas from classical conditioning. The approach to analysing classical conditioning begins with a derivation of a list of criteria of what a passive learning system would need to be able to do in order to learn a model of its environment, regardless of any function, implementation, or approach. These criteria give a context in which classical conditioning can be analysed.

Classical conditioning itself consists of many different phenomena, as shall be discussed in the first section of this chapter. Therefore, in the analysis of classical conditioning, the form it will take will be to introduce each of the phenomena in turn and then compare each individual phenomenon against the criteria that were previously derived. The list of phenomena covered by the analysis is larger than those that could be feasibly implemented in time allowed for a single project. Because of this, the list of phenomena is split in two – those phenomena that are used by the system introduced in chapter four, and those that are not. It is however still important to cover the phenomena that are not used by the system, as it is only when they are included that the correspondence between the criteria derived and the phenomena of classical conditioning can be demonstrated.

Analysing classical conditioning in this manner demonstrates that it can satisfy all of the criteria without reference to any underlying learning mechanism. This gives a basis to both hypotheses. Firstly, a basis is given to hypothesis A by showing that classical conditioning can be used as a machine learning specification. Secondly, there is a basis for hypothesis B because the analysis is based on a passive learning system.

Due to the analytical process described above, the structure of this chapter is as follows: The first section will give a short, abstract overview of classical conditioning, introducing the basic concepts and terminology. Also included in this first section are some general analytic remarks about classical conditioning that need to be made, but are linked neither to the criteria nor any specific phenomenon. After that basic overview and analysis, the second section derives and presents the twelve criteria that the phenomena of classical conditioning are compared against. The next two sections introduce and analyse twenty-eight of the phenomena of classical conditioning. The first of these two sections covers those phenomena that are directly used by the system presented in chapter four and the section after covers those that were not. After the phenomena are introduced and analysed, the final section discusses two competing interpretations of classical conditioning from within the classical conditioning community, as the work of this thesis is based upon one of these interpretations.

2.1 An Overview of Classical Conditioning

Classical conditioning is the name given to a collection of psychological phenomena that are concerned with associative learning. These phenomena are also known as Pavlovian conditioning, named after Ivan Pavlov, one of the primary people who introduced the theory. Pavlov's widely-known experiments with dogs, first published in English in 1927 (Pavlov, 1927) were among the first experiments to demonstrate the fundamental phenomena of Classical Conditioning. Pavlov's cardinal experiment was to create an audible tone (mostly a bell or metronome) immediately prior to the dogs having a substance directly placed into their mouth that would cause the reflex action of salivation (usually meat powder or a weak acid). This was done multiple times. The same audible tone was then presented to the dogs without the presentation of the substance. The result was that the dogs' salivary response was observable with the tone even when substance was not presented. The extent of the salivary response without the substance correlated with the number of prior presentations where both the tone and substance were presented jointly. Pavlov used this experiment and others like it to derive a theory of learning.

The theory of learning that Pavlov derived from this is that an arbitrary neutral stimulus can become associated with any non-neutral stimulus, (i.e. a stimulus that triggers a reflex response) based on their similar co-occurrence in time. Thus when the neutral stimulus is presented alone, the subject gives a related response to the unconditioned response. In the literature surrounding classical conditioning, the names of the stimulus and the responses have particular names. The neutral stimulus is known as the conditioned stimulus (C.S.) which in Pavlov's experiment corresponds to the generated tone. The non-neutral stimulus is termed the unconditioned stimulus (U.S.) which in Pavlov's experiment corresponds to the substance placed in the dogs' mouths.

The response to the non-neutral stimulus is called the unconditioned response (U.R.) which in Pavlov's experiment corresponds to the salivary reflex the dogs had to the substance. The response to the neutral stimulus after the association had been formed is the conditioned response (C.R.) which in Pavlov's experiment corresponds to the salivary response the dogs had to the tone when the substance was not present.

There are five general observations that have arisen as a consequence of the analysis of classical conditioning. The first and most important of which is that classical conditioning can be seen as a mechanism to learn predictions. This is not a unique insight of itself, for example the S.B. model (Sutton & Barto, 1981), a model of classical conditioning presented in chapter three, is based on such an observation. However, no mention has been encountered in the classical conditioning literature discussing the issue of noise within the learning process. Noise and the dangers of over-fitting are central concepts within the topic of machine learning (Russell & Norvig, 2003, p. 657 & p. 662) and a key component of the analysis presented in this chapter.

The second observation is that different approaches to understanding the process of learning have given different names to what can be argued is the same thing. In the section deriving the criteria of a learning agent, it is called a sensor state. In the classical conditioning literature, it is called a stimulus. In the artificial intelligence literature it is called an event. Each of these terms makes sense when used within their particular approach; sensors convey their information in series of momentary states; an animal's neurones are stimulated by a stimulus; an observed event can predict another event. However, when combining the approaches, this can become confusing. Therefore from this point on in this thesis, unless the subject of discussion is better served by using a particular term, this thesis shall refer to all three as an event. The reason the term "event" is chosen it that it is the most general term in a semantic sense, but from the point of view of an agent, all three can be said to be equivalent. Momentary input states can be said to be the smallest perceptible component of a stimulus because their current state is caused by some stimulus and they can be said to be events because they are caused by a dynamic environment and each momentary state value has a definite start and end time. Stimuli can be said to be a set of momentary input states because their presentation causes a set of those states to change and they can be said to be events because they are presented to the subject for a definite period of time. Events can be said to be reducible to input states from the subjective view of an agent because events are observed as changes in input states, and events can be said to be stimuli because an event cannot be observed if it does not stimulate some part of an agent.

The third general observation is that, from a logical semantics point of view, conditioning can be seen as a process of discovering material implications between two events spread over time. This can be seen in the requirement for both congruity and contingency between a first event and a second. Because of

sensory preconditioning and secondary conditioning, phenomena of classical conditioning that shall be discussed later in this chapter, this effect can be seen to be independent of the need for an unconditioned stimulus for these implications to form.

The fourth general observation is that the magnitude (or salience / noticeability) of an event is a factor in the conditioning process. This can be seen to be the strategy employed to deal with the issue of having more sense data available than can be processed in the time allowed (criterion 11 of the criteria derived in the next section). This is a rational strategy as by prioritising the data in order of magnitude the agent can process the data in the likely order that an event will have a large effect on the environment and/or the agent.

The fifth and final general point is that the amount of response given in anticipation of an event can be seen to be a gradual change proportional to the certainty of the predicted event and its expected magnitude. The anticipation also gradually changes inversely proportional to the expected timing of the predicted event. This gradual response allows for an optimal use of the resources used in the response (criterion 12 of the criteria derived in the next section). By responding proportionally to the certainty, the agent is hedging its bets that the predicted event will occur. By responding proportionally to the expected magnitude of the predicted event, the agent is using as much resources as needed to deal with the predicted event and no more. By responding in an inversely proportional manner to the expected timing of the predicted event, the agent is avoiding committing to sustaining a large quantity of resource over a long time without any return on that investment.

2.2 An Analysis of the Criteria for a Passive Learning System

Both hypothesis A and hypothesis B refer to the need for an agent to build a model of its environment. This section attempts to derive a list of criteria for a passive learning system from first principles, independent of any conception about how a criterion might be satisfied by a learning system. As it is from first principles, some of the arguments may seem obvious – this is due to attempting to be comprehensive in the derivation of the criteria. This is a list of criteria assuming that the agent is learning in a passive manner – where the agent is not able, through its actions, to influence the state of the environment. However, some considerations about the needs of an active agent are noted within the argument, as they have some bearing on the analysis of the classical conditioning phenomena presented in the next two sections.

The purpose of modelling any form of phenomenon or environment is that it allows for the user of that model to attempt to predict the future given the current state. By imbuing an agent with the capacity to create an internal model of its environment, it allows the agent to predict the imminent behaviour of that environment and so allow a rational response to that behaviour.

The only way an agent can build a model of its environment is through its senses. This means that for any agent, before any form of inference is applied, all that exists is a great many typed input signals each providing data of a particular momentary value. Through the use of memory, those momentary signals can be combined with some timing data. Through the signal's data type, those momentary signals can be augmented with both its spatial relationship to other signals and the nature of the input (e.g. the signal of a visual sensor or a microphone input). This gives the agent a many-channelled stream of typed input values from which a model must be built. Therefore, as this is the only data available to the agent, by the definition of a model, the task of creating an agent that is able to learn a model of its environment is the same as the task of building an agent with the ability to predict how a given signal will change based on the current and previous states of all the signals available to that agent.

In any consistent environment, if the same type of event happens more than once, at least some of the signals will repeat the same pattern of input signals. These patterns of signal values could be repetitions across the spatial, temporal or source-type dimensions, or any combination of those dimensions. It is through these repeated patterns that a learning system must learn to create its prediction – by coming to expect what comes next in the patterns. Stated in logical terms, the patterns that need to be learned are chains of signal states implying that another signal state will follow. Therefore, any learning system will need to learn its environmental model from repeated co-occurrence of particular signal values. It will also need to allow for the chaining of these predictive states to allow for more complex patterns.

Some patterns of signals will be independent of the magnitude of those signals. In other words, the same pattern holds even though the signal values themselves have been subjected to the same function – for example a pattern of signals has been subjected to addition or multiplication by a constant factor. An example of this that a pattern signal increases in value three times in a row followed by a reduction in value three times in a row, this pattern would hold whether the interval of each increase is 1 or a 100. An example of this can be seen in the colour constancy effect (Mollon, 1985); the same object can appear to be different colours under different lighting yet a human can still identify what the object's colour would be under white light. This implies the need for a passive learning system to recognise a pattern independently of variations in the magnitude of the pattern. However, there needs to be a limit to the variations allowed. Otherwise, allowing sufficient changes in the magnitude of a pattern would allow that pattern to match in circumstances where it does not apply, especially if the magnitude changes are non-linear. All this implies that a criterion for a passive learning system is that the system should allow for bounded variance within a pattern.

When attempting to predict a given signal state value, there will be several candidate signal states that have co-occurred to greater or lesser extents. Some

of these signal states will be due to mere happenstance, some due to a true environmental pattern. The latter are the desired pattern states the former are termed noise and should be ignored. Therefore another requirement for a learning system is to separate the true predictive states from the noise states.

The need to separate noise from true predictions leads to two trade-offs. The first of these trade-offs is between reliability of prediction and the speed at which a prediction is learned. In an ideal world, a learning system would instantly learn patterns of events based upon one presentation – this would allow for rapid learning of the environmental model. However, due to the presence of noise, more instances of a pattern are needed to ensure that an observed pattern is not just noise. Note that it could be the case that the same happenstance occurred a second or subsequent time. All this entails that a learning system needs to be able to sensibly deal with the trade-off between the reliability of a prediction and how quickly that prediction is acquired.

The second trade-off is between the reliability of a prediction and how far in advance a prediction can be made. The further in advance a prediction can be made, the more time the agent has to make any preparations for any particularly desirable or undesirable signal states. However, the further in advance an agent attempts to predict a state, the more potential predictive states have to be evaluated, as more time has passed for there to be more variety of signals to occur. This wider selection of signal states allows for a greater chance for a noisy co-occurrence to recur, which could allow for the noise signal to be seen as a predictor. Therefore a learning system needs to be able to sensibly deal with the trade-off between the reliability of a prediction and how far in advance that prediction can be made.

Because of the previous two trade-offs, if an agent is particularly unlucky in its experience of noise, then it may learn an incorrect prediction due to that noise. The agent thus needs the capacity to undo that learned prediction in the face of new evidence that it is untrue. There is also another case where a prediction made by the model would need to be retracted. In a dynamic environment, patterns that were once present can disappear over time, simply due to the environment changing around the agent. These trade-offs also need to be dealt with in the case that the pattern is dependent on a random occurrence. This will be discussed later in this section.

There is another consequence of a dynamic environment. Some behaviours of an environment can be state-dependent. What is meant by state-dependent is that a pattern of observed signal values may be dependent on hidden or non-local aspects of the environment. A more concrete example of this would be a sequence of lights that follows different patterns depending on the state of a hidden switch. In other words, there is a class of patterns that can be present some of the time but only in the correct circumstances. These circumstances may or may not have a direct signal or set of signals that indicate whether a particular pattern will be able to be detected or not. There may be signal indicating a pattern is able to be found or a signal that indicates that a pattern

may not be found. These two cases are independent of each other, leading to four distinct cases that a learning agent needs to be able to learn within. The first case is that two signals or sets of signals exist, one that indicate that the state-contingent pattern may be able to be found and another that indicate when that pattern will not be found. The second case is that there exists a signal or set of signals that indicate that a state-contingent pattern may be able to be found but there does not exist a signal or set of signals that indicate that that pattern may not be able to be found. The third case is that there does not exist a signal or set of signals that indicate a pattern may be able to be found but there does exist a signal or set of signals that indicate a pattern will not be found. The final case is that no signal or set of signals exists that either indicate that a pattern is able to be found or is not able to be found. All four of these cases need to be dealt with by the agent. Table 2.1 summaries these states.

Case	**The pattern is able to be found**	**The pattern is not able to be found**
1	Indicator signal exists	Indicator signal exists
2	Indicator signal exists	No indicator signal exists
3	No indicator signal exists	Indicator signal exists
4	No indicator signal exists	No indicator signal exists

Table 2.1: A summary of the four situations a state-dependent pattern can be in.

Some signal states do predict other signal sates, but in a stochastic or indeterminate manner rather than in a directly deterministic manner. These patterns of prediction should still be learned by the agent as the pattern does exist. However, this complicates the need to avoid noise, as it becomes harder to distinguish happenstance from a genuine probabilistic relationship. Even a single pairing of two events can be seen as evidence of a very low probability stochastic relationship. Therefore a learning system needs to be able to learn stochastic patterns of signal states while sensibly dealing with the trade-off of discriminating between stochastic patterns and noise.

The learning system needs to fulfil all of the previous needs in a computationally efficient manner. Without designs and compromises in favour of computational efficiency, the task of learning new patterns and even using any existing learned patterns can become slow to the point where an active agent is no longer able to react to the environment timely enough. With this in mind, a learning system needs to have strategies in place to address this compromise between the breadth and depth of the patterns the agent can learn and computational efficiency. For systems that have a great number of different channels of sense data, this compromise may lead to some sense data not being processed in any meaningful way. Due to this a learning system will need to have a strategy to deal with having more sense data than can be realistically processed.

Finally, some acknowledgement is required for how a passive learning system behaves as part of a wider active learning system. This limited acknowledgement is required to allow for a more complete interpretation of the phenomena of classical conditioning as it learns passively – as it is the case that any passive learning in classical conditioning is a part of a wider active learning system. If a passive learning system is a part of a wider agent that acts within its environment then this may have extra consequences for the passive learning system – as is touched upon with earlier parts of this discussion. Reacting to a particularly desirable or undesirable sensor state may also use up resources available to the agent – such as energy. Therefore a passive learning system that is a part of an active learning agent needs to learn in a way such that the predictions it learns allow for optimal use of the agent's resources.

The following list summarises the issues a passive learning agent will need to deal with that have arisen from the above discussion:

1. A passive learning system needs to learn its environmental model from repeated co-occurrence of particular signal values.

2. A passive learning system needs to allow for bounded variance within a pattern.

3. A passive learning system needs to allow for the chaining of predictive states to allow for more complex patterns.

4. A passive learning system needs to separate the true predictive states from the noise states.

5. A passive learning system needs to sensibly deal with the trade-off between the reliability of a prediction and how quickly that prediction is acquired.

6. A passive learning system needs to sensibly deal with the trade-off between the reliability of a prediction and how far in advance that prediction can be made.

7. A passive learning system needs to undo a learned prediction in the face of new evidence that indicates that the prediction is untrue.

8. A passive learning system needs to learn that a given pattern of predictions may be contingent on the environment being in the correct state. This contingency needs to be learned where:

   (a) There is a signal state or set of signal states that indicate when a pattern can be found and also a signal state or set of signal states that indicate when that pattern cannot be found.

   (b) There is a signal state or set of signal states that indicate when a pattern can be found but not a signal state or set of signal states that indicate when that pattern cannot be found.

(c) There is not a signal state or set of signal states that indicate when a pattern can be found but there is a signal state or set of signal states that indicate when that pattern cannot be found.

(d) There is not a signal state or set of signal states that indicate when a pattern can be found and also there is no signal state or set of signal states that indicate when that pattern cannot be found.

9. A passive learning system needs to learn stochastic patterns of signal states while sensibly dealing with the trade-off of discriminating between stochastic patterns and noise.

10. A passive learning system needs to have strategies in place to address the compromise between the breadth and depth of the patterns the agent can learn and computational efficiency with which they are learned.

11. A passive learning system needs to have strategies to deal with the case of having more sense data than can realistically be processed.

12. A passive learning system that is a part of an active learning agent needs to learn in a way such that the predictions it learns allow for optimal use of the agent's resources.

2.3 Phenomena Used by the System

Since Pavlov's initial discovery of the learned association between stimuli, a wide range of connected phenomena concerning the interaction of conditioned stimuli and unconditioned stimuli have been discovered. The main phenomena are described and analysed in this and the next section. Schmajuk (2008) provides a comprehensive yet concise overview of most of the phenomena in both sections.

As previously discussed, the phenomena are primarily analysed against the criteria derived in the previous section. The analysis parts of this and the next section demonstrate that classical conditioning satisfies all of the criteria, lending support to both hypotheses.

The split between the phenomena in this section and those in the next section is to some extent quite arbitrary, being sorted by whether a phenomenon has been explicitly used by the system described in chapter four. However, this section does include the phenomena that are arguably the most fundamental, such as acquisition and extinction and those that are the most discussed in the literature, such as blocking and conditioned inhibition. This is because it is quite natural to prioritise the implementation of these phenomena when building the system presented in chapter four. This does not however degrade the importance of those phenomena included in the next section.

To summarise, the phenomena described and analysed in this section are:

1. Acquisition
2. Extinction
3. The Inter-Stimulus Interval
4. Reacquisition
5. Blocking
6. Recovery from Blocking
7. Conditioned Inhibition
8. Extinction of Conditioned Inhibition
9. Latent Inhibition
10. U.S.-Pre-Exposure Effect
11. Sensory Preconditioning
12. Secondary Conditioning

2.3.1 Acquisition

Acquisition is the process whereby the conditioned stimulus becomes associated with the unconditioned stimulus and thus the conditioned response. This is the phenomenon that was discovered first, as discussed in section 2.1. The strength of the association (e.g. measured by the amount of saliva produced) is a sigmoid-like function of the number of reinforcements of the conditioned stimulus (i.e. the number of presentations of the conditioned stimulus where the unconditioned stimulus follows). Acquisition was first described by Pavlov (1927).

2.3.1.1 Analysis

Almost by definition, acquisition demonstrates learning from repeated co-occurrence of particular events (criterion 1). However, the phenomenon also helps fulfil other criteria in the way those co-occurrences are used. The process indicates that the agent only needs an iteratively updated current measure of association between any two events that have previously been associated, as opposed to the storage of the complete association history of each pair of events. By only using a current measure, the computational costs of storing that data are minimal for each pair of events (criterion 10).

The use of a sigmoid-like curve offers further contributions towards satisfying the identified criteria. This can be seen by comparing the sigmoid curve against a linear curve. The sigmoid initially accumulates lower association strength than the linear curve, followed by a sudden fast increase in accumulation giving greater association strength than linear, and followed by a tail-off as the association strength asymptotically approaches one. This pattern of accumulation can be seen to be acting as a "soft" threshold, where there is always an increase, but has almost all of its gains around the threshold value. This allows for the suppression of noise (criterion 4), as noise is likely to occur at a lower frequency than true event patterns.

2.3.2 Extinction

Extinction is the process whereby a conditioned stimulus that is already associated with the unconditioned stimulus is repeatedly and consistently presented to the subject without the unconditioned stimulus. The strength of the association is weakened and eventually returns to the same level of association as observed prior to acquisition. Extinction was first described by Pavlov (1927).

2.3.2.1 Analysis

Like acquisition, extinction is almost by definition a mechanism that can undo a learned prediction in the face of new evidence that indicates that the prediction is untrue (criterion 7). Again, there is some contribution to other criteria too. Extinction works at all association strength levels, not just at high levels. This means, that when combined with the initial slow accumulation of association strength, extinction can help keep noise below the soft threshold of the sigmoid curve (criterion 4).

2.3.3 The Inter-Stimulus Interval

The inter-stimulus interval (I.S.I.) is the time between the start of the conditioned stimulus and the start of the unconditioned stimulus. This interval induces two modes of acquisition: delay and trace conditioning. Delay conditioning is where the conditioned stimulus overlaps or finishes immediately before the unconditioned stimulus appears. Trace conditioning is where the conditioned stimulus finishes with a period of inactivity before the unconditioned stimulus appears. The inter-stimulus interval affects the rate of acquisition of a C.S.-U.S. association. The rate follows a curve where small intervals are negligible, it then rapidly moves up to a peak and then gently decays, similar to the curve of a log-normal distribution. An idealised version of this is shown in Figure 2.1. The difference between delay and trace conditioning is that the latter has a much faster decay after the peak. The effect of the inter-stimulus interval was first described by Pavlov (1927). The shape of the curve was built up over a number of studies, notably by Schneiderman & Gormezano (1964) and by Smith (1968).

2.3.3.1 Analysis

When attempting to learn with long intervals between the predictor event and the predicted event, there needs to be a cut-off point after which a prediction is not processed. The reason this is needed is that otherwise, every event the learning system has ever experienced would need to be associated to some extent with every other event type that has ever been experienced. Not only is this computationally inefficient, violating criterion 10, it makes the learned associations become useless, as every event would to some extent predict every other event that came after it in the entire lifetime of the agent.

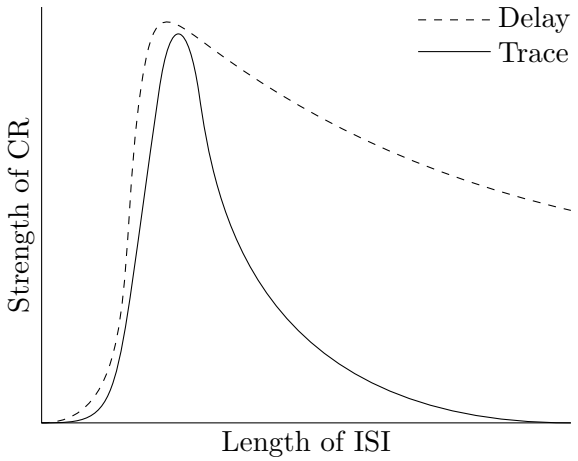

Figure 2.1: An idealised diagram showing the effects of the I.S.I. for a fixed number of reinforcements.

By reducing the association strength as the inter-stimulus interval increases, it implies that at some point the gain in association strength from a pairing of two events will drop to nothing (assuming that association strength is quantised at some level). This gives a natural cut-off for computational efficiency (criteria 10 and 11).

The gradual reduction of the association strength gains as the inter-stimulus interval increases also allows for a sensible trade-off between the reliability of a prediction and how far in advance a prediction can be made (criterion 7). The gradual reduction means that noise events with a high inter-stimulus interval have less influence. This takes into account the fact that it will be likely that more noise events occur for longer inter-stimulus intervals.

When a preceding event does not finish as or after the predicted event starts, as is the case with trace conditioning, there is a higher likelihood that the event is noise rather than being a proper predictive relationship. This can be seen to be because of a poorer temporal relationship with the predicted event. Therefore trace conditioning implies a higher likelihood that a given event is noise, for a fixed inter-stimulus interval. Therefore, the faster drop-off of gains in association strength in the case of trace conditioning allows the learning system to retain a sensible trade-off between the reliability of a prediction and how far in advance a prediction can be made (criterion 6).

The reduction in association strength gains for longer inter-stimulus intervals also allows for a better allocation of resources, as low association strengths found with long inter-stimulus intervals elicit only a weak response. This allows the total resources used by maintaining a ready state over a long time to be minimised (criterion 12).

Predictive value also serves a possible explanation for the reduction in association strength gains for very short inter-stimulus intervals, as for very short intervals, the prediction is too immediate to be useful. In addition, as the difference between the start points of each event decreases, it becomes preferable to see the two events as being part of a parallel compound event rather than having a predictive relationship.

2.3.4 Reacquisition

Reacquisition is the name given to the phenomenon where a previously extinguished C.S.-U.S. association is acquired again. During reacquisition, it takes fewer reinforcements to re-acquire the same strength of association than it did the previous time that the association was acquired. Reacquisition was first described by Pavlov (1927). Figure 2.2 exhibits an idealised version of how reacquisition (and by extension, acquisition and extinction) modifies the strength of the conditioned response based on the number of presentations of the relevant stimuli. The curve shows four cycles with each cycle split into two phases. The first phase reinforces the conditioned stimulus (labelled by "R") and the second phase non-reinforces the conditioned stimulus (labelled by "NR"). Each "R" phase shows the sigmoid-shaped curve of the acquisition phenomenon. Each "NR" phase shows the sharp linear decay of the extinction phenomenon. Each phase presents the same number of relevant stimuli. For each new cycle, the number of presentations required to reach the peak conditioned response strength of the previous cycle is lower. This shows the phenomenon of reacquisition. The figure is based on the results presented in Balkenius and Morén (1998).

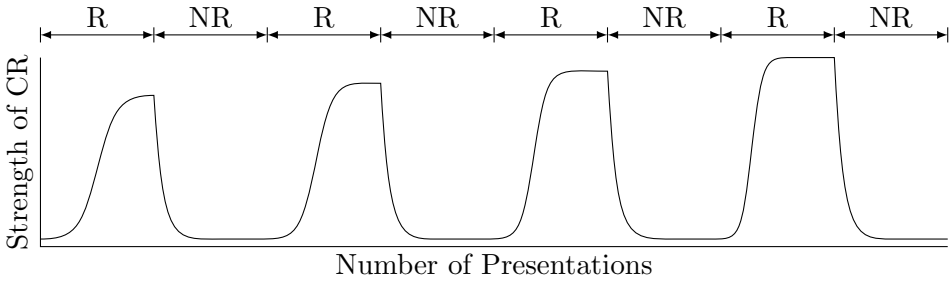

Figure 2.2: An idealised diagram showing the effect of Reacquisition.

2.3.4.1 Analysis

Consider an environment where a part of that environment is state-dependent, but there may or may not be any cues to give any indication which state that part of the environment is currently in (criterion 8 a–d). To be able to adapt to the case where an association between events gets observed in intermittent bursts, the agent would need to learn to quickly switch between a state where the association exists and a state where it does not exist.

The process of reacquisition can show how this can be done within the paradigm of conditioning. By allowing gradual speed-up of the rate of acquisition, the switching between states of the association existing will eventually allow for full or near-full acquisition of association strength within a single observation, and the same for extinction. This allows for rapid switching between states of an association existing and not existing.

This ability to switch should only be gained gradually, with each switch between states of strong association requiring fewer observations. If this were

not the case, then after acquiring a strong association for the first time, any extinctions and further acquisitions would cause the agent to switch between high association strength and low association strength. This is undesirable because such a system would violate criterion 3: If there was an initial glut of the same form of noise, such that a high association strength value is attained, but later became extinguished due to the nature of it being noise, then every instance of that type of noise would later cause the full reinstatement of that association. By gradually reducing the number of instances required for the reinstatement of a strong association it means that single occasional instances of a past noise-based strong association does not fully reinstate that strong association. While a single noisy stimulus paring will still be associated, this noise will more likely be extinguished quickly, and so the acquisition rate will only increase by an insignificant amount.

2.3.5 Blocking

Firstly the subject is conditioned to a C.S.-U.S. pairing. Secondly, another conditioned stimulus is introduced alongside the original pairing (such that all three stimuli are present) and provided with further reinforcements. The second conditioned stimulus will only ever display a weaker response on its own than would normally be the case for the number of reinforcements. This is known as the blocking effect and was discovered by Kamin (1969).

2.3.5.1 Analysis

The blocking effect only occurs when both potential predictor events are present. If both potential predictor events are presented as separate trials, then both associations can become strong associations. From this, it can be seen that the reason for the blocking effect is that it allows for noise events to occur at the same time as a predictor event and its corresponding predicted event. In the case where there is an event with a strong association to the predicted event and another event which has little or no association strength, the strong association is able to adequately predict the predicted event on its own, and so little credit can be given to the event with little or no association strength. This allows the learning agent an additional robustness to noise (criterion 4).

2.3.6 Recovery from Blocking

Blocking only holds while a strong C.S.-U.S. association holds. If the strong association is extinguished, then it will no longer prevent any other conditioned stimulus from having a conditioned response due to an association with the unconditioned stimulus. This effect was found by Matzel *et al.* (1985).

2.3.6.1 Analysis

As described above, blocking can be seen as a mechanism where the presence of an event that is adequately predictive, can stop another (as yet unseen) event from becoming a predictor of the same event. If that predictor event ceases to be a good predictor through extinction, then there is no reason to stop another event from becoming a predictor. Because of this argument, this phenomenon can be seen as another strategy to separate out predictive events from noise events (criterion 4), albeit from a reverse approach to the blocking effect.

2.3.7 Conditioned Inhibition

Conditioned inhibition refers to an effect where the presence of a specially conditioned stimulus can reduce the response of a different conditioned stimulus when it is present (with little or no effect when the stimulus is not present). This can be demonstrated in the following experiment: First, two conditioned stimuli are conditioned separately to associate with the unconditioned stimulus. A third stimulus is then presented simultaneously with one of the two previously reinforced conditioned stimuli without the unconditioned stimulus. After repeated trials of this presentation, presenting this third stimulus along with the other of the two conditioned stimuli will not elicit a conditioned response. When either of the conditioned stimuli is presented without the third stimulus, the conditioned response is elicited. This effect was first described by Pavlov (1927) and later expanded on by Rescorla (1969).

2.3.7.1 Analysis

Conditioned inhibition can be seen as a mechanism to allow an agent to learn that the presence of an event indicates that a particular event pattern will not be able to be found. In this way, conditioned inhibition can contribute to the learning of state-dependent patterns of events. More specifically, conditioned inhibition can assist learning a pattern of events that is contingent on an environmental state in the cases where there exists an event that indicates that a particular pattern cannot be found (criterion 8a and criterion 8c).

2.3.8 Extinction of Conditioned Inhibition

Analogous to the extinction effect, any learned inhibitory effect of a stimulus can be reversed. This is done by presenting the inhibitory stimulus together with the unconditioned stimulus that the inhibitory stimulus inhibits. This effect was first described by Pavlov (1927).

2.3.8.1 Analysis

As the conditioned inhibition mechanism is iterative, a conditioned inhibitor could be created through happenstance and therefore needs to be removed.

Another situation where a conditioned inhibitor needs to be retracted is where a conditioned inhibitor that was true in the past has ceased to be true due to a change in the environment. Extinction of conditioned inhibition demonstrates that such a mechanism exists. This helps a learner retract predictions (in this case a prediction of the non-occurrence of an event) that are shown to be untrue in the face of new evidence (criterion 7). As inhibition associations are gained as a gradual process like acquisition, this extinction process can help stop incorrect inhibitory associations from forming (criterion 4), by reducing the strength of any weak inhibitory association.

2.3.9 Latent Inhibition

When a stimulus is exposed to the subject prior to it being reinforced with the unconditioned stimulus to form an association, the conditioned response of the stimulus is weaker than would be the case if the subject was not pre-exposed. This phenomenon is called latent inhibition and was discovered by Lubow and Moore (1959).

2.3.9.1 Analysis

In the discussion of the contingency phenomenon in section 2.4.1, latent inhibition is involved. In that discussion it is likened in its effect to that of extinction. Latent inhibition could be seen as a form of pre-emptive extinction, reducing the predictive value of the event to future new associations, as it can be assumed that it is a noisy association because the effect was not observed initially. From this interpretation, latent inhibition can be seen to be a strategy that allows a learning system to be more resilient to noise (criterion 4). By still allowing association strength to build, but at a slower pace, new associations can still be made but at a pace that would cause noise to not build association strength.

2.3.10 U.S.-Pre-Exposure Effect

The U.S.-pre-exposure effect is a duel phenomenon with latent inhibition. When the unconditioned stimulus is exposed to the subject without a conditioned stimulus and then later is exposed with a conditioned stimulus, the conditioned response is weaker than would be the case if the subject was not pre-exposed to the unconditioned stimulus. This phenomenon was also discovered by Lubow and Moore (1959).

2.3.10.1 Analysis

The U.S.-pre-exposure effect can be seen to be the same effect as latent inhibition but for the predicted event of an associative event pair, as was briefly mentioned in the discussion of the contingency phenomenon. By observing the event to be predicted prior to any associations, the predictive value of future potential predictor events is damaged, as those future potential predictor

events did not predict the initial observations of the event to be predicted. As with latent inhibition, the purpose of this can be seen to be another strategy to allow the learning system deal with noise (criterion 4). Again, this is done in a way that new predictors are allowed, but they have to be proved to be a predictor to a greater extent before they can be bestowed with a strong association.

2.3.11 Sensory Preconditioning

Sensory preconditioning involves two stages. In the first stage, two neutral stimuli are presented together. In the second stage, one of those stimuli is paired with an unconditioned stimulus. Presentation of the stimulus that was not paired with the unconditioned stimulus still produces a conditioned response, despite not ever being presented alongside the unconditioned stimulus. Brogden (1939) first demonstrated this effect.

2.3.11.1 Analysis

Both sensory preconditioning and secondary conditioning (discussed below) are experimental results that indicate the same root phenomenon, namely that classical conditioning allows for chaining of predictive events (criterion 3). By chaining predictive events, a learning system is able to build up a model that allows for prediction of future events further in advance than would be the case if chaining did not happen. This allows for a learning system to increase how far in advance predictions can be made in a way that maintains some level of reliability (criterion 6) because it gives intermediate predictions.

2.3.12 Secondary Conditioning

Also known as second-order conditioning and can be seen as a phenomenon related to sensory preconditioning. Secondary conditioning is where a secondary conditioned stimulus can be conditioned to elicit a conditioned response through reinforcement only with a primary conditioned stimulus (where a primary conditioned stimulus is one that has been reinforced with the unconditioned stimulus). This effect is typically weak as the extinction of the primary conditioned stimulus will happen while the secondary conditioned stimulus is being conditioned. This effect is highly dependent on the saliency of the unconditioned stimulus. Tertiary conditioning, where there is a chain of three unconditioned stimuli, can occur but is usually too weak to observe, unless the unconditioned stimulus is particularly salient – for example extreme pain. Along with sensory preconditioning, Brogden (1939) also found this effect.

2.3.12.1 Analysis

As described in the discussion of sensory preconditioning, both phenomena can be seen to be results indicating that predictive events can be chained

(criterion 3) and allow for a way to increase how far in advance predictions can be made (criterion 6). With secondary conditioning there is a need to also look at the related results of tertiary conditioning – as its results could indicate that chaining is a limited effect. Arguably, instead of tertiary conditioning occurring rarely, tertiary chaining (and higher-order chaining) happens very regularly. In this alternative view, there is a two-fold cause of the low measured conditioned response. Firstly, the extinction effect is also occurring while doing the trials for both secondary and tertiary conditioning.

Another potential reason for a reduction in measurable response is that when there is an increase in the expected timing of an event that requires a response. The need to use resources to pre-emptively respond to that expected event becomes less pressing, and so in order to optimise use of finite resources, a response is less forthcoming for higher-order chains of conditioning – this can be seen as a strategy to allow for the optimal use of resources (criterion 12).

2.4 Phenomena Not Used by the System

While reviewing the literature of classical conditioning, a total of twenty-eight different phenomena were found. Of those phenomena, there was only enough time to directly implement twelve. However, the remaining sixteen are still worth discussing. This is because when the respective analyses of all the phenomena presented in both sections is taken into account, it can be seen that classical conditioning as a whole satisfies the criteria derived in section 2.2.

The phenomena covered in this section are:

1. Contingency
2. Generalisation
3. Discrimination
4. Configural Cues
5. Patterning
6. The Inter-Trial Interval
7. Facilitation of Remote Associations
8. Primacy
9. Temporal Primacy
10. Learning-to-Learn
11. Overshadowing
12. Super-Conditioning
13. Backward Blocking
14. Partial Reinforcement
15. Partial Reinforcement Extinction Effect
16. Spontaneous Recovery

2.4.1 Contingency

While the majority of the phenomena discovered deal with the congruity of stimulus events (i.e. the events occur in the same time and space), congruity is necessary but not sufficient. For conditioning to occur, the conditioned stimulus also needs to probabilistically predict when the unconditioned stimulus occurs. This means that the probability of the occurrence of the unconditioned stimulus given that the conditioned stimulus did occur must be greater than the probability of the occurrence of the unconditioned stimulus given that the conditioned stimulus did not occur. Stated symbolically: $P(US.|C.S.) > P(US.|\overline{C.S.})$. The requirement for contingency was experimentally confirmed by Rescorla (1968).

2.4.1.1 Analysis

By requiring there to be a dependency in the Bayesian sense between two events for a strong association to occur, it allows for a greater resistance to noisy predictions (criterion 4). This is because it allows for a measure of the absolute probabilities of each event occurring to influence the estimate of the events co-occurring. This use of an absolute probability of each event occurring also contributes to the ability for the learning system to be able to retract predictions in the face of new evidence – as it implies that each presentation of either the predictor event or the predicted event without the other will reduce the association strength (criterion 7).

This thesis holds that this phenomenon of contingency is mostly an epiphenomenon that arises from some of the other phenomena discussed by this thesis. To see this, consider that this phenomenon is stating that the association strength between the predictor event A (the conditioned stimulus) and the event to be predicted B (the unconditioned stimulus) is proportional to the difference between the probability that B will happen given that A has happened and the probability that B will happen given that A has not happened. This can be expressed in the derivation expressed in equations 2.1 to 2.3, assuming that the association strength between A and B is $V(AB)$, and then applying the definition of conditional probability.

$$V(A,B) \propto P(B|A) - P\left(B|\overline{A}\right) \tag{2.1}$$

$$\propto \frac{P(A \cap B)}{P(A)} - \frac{P\left(\overline{A} \cap B\right)}{P\left(\overline{A}\right)} \tag{2.2}$$

$$\propto \frac{P(A \cap B)}{P(A)} - \frac{P(B) - P(A \cap B)}{1 - P(A)} \tag{2.3}$$

The final expression implies that any value that is proportional (or inversely proportional) to at least one of the three values of $P(A)$, $P(B)$ and $P(A \cap B)$ can be used as part of a calculation that produces a value that is proportional to the association strength $V(AB)$. This thesis argues that the

other phenomena do produce such values. When acquisition occurs, all three values increase, as positive instances of all three are observed. When extinction occurs, $P(A)$ increases but the other two values do not change, as only instances of A are observed; the same is the case for latent inhibition. When the U.S. pre-exposure effect occurs, $P(B)$ increases but the other two values do not change, as only instances of B are observed.

Therefore this phenomenon arguably mostly happens as a consequence of the other observed phenomena. The reason the qualification "mostly" is applied is that the results of this phenomenon does imply that observation of event B (the unconditioned stimulus) will decrease the association strength after the first co-occurrence of the two events together too, as an analogue to the extinction phenomenon. If such a phenomenon is observed, then this phenomenon can be said to be just a consequence of other phenomena (or, conversely, the other phenomena are a consequence of this one phenomenon). Note that this analysis is expanded in the future work section of chapter seven, in section 7.3.14 and this expansion indicates that extinction due to the presentation of B-only instances may only occur in young subjects.

It should also be noted that the classical conditioning literature considers contingency to be a prerequisite of conditioning rather than a phenomenon of conditioning. This thesis finds this distinction to be arbitrary and therefore redundant. This is for two reasons: Firstly, from a semantic perspective, the term "phenomenon" is a general enough that it should include "prerequisite". On a more fundamental level, if the previous analysis is correct, then contingency is not a prerequisite that learning arises from at all, but is instead an epiphenomenon that arises from learning. If it is indeed a prerequisite, then there needs to be an explanation for how the subject detects that that the prerequisite is in place in the first instance, so that learning is allowed to occur.

2.4.2 Generalisation

A first conditioned stimulus is reinforced with the unconditioned stimulus. If a new stimulus is then presented that shares enough characteristics with the first conditioned stimulus, then a conditioned response is elicited. This is despite the fact that the subject has never been exposed to that stimulus. This effect was first described by Pavlov (1927).

2.4.2.1 Analysis

Generalisation is based on the assumption that if something has a similar appearance, it may have similar behaviour. This can be a reasonable assumption, for instance when encountering different members of the same species. The use of the assumption helps learning in three ways. Firstly, it allows the learner to have some tolerance to the variance in the events that are recognised (criterion 2).

Secondly, it offers a reasonably reliable short-cut to allow a learned prediction to be used when faced with a novel, but similar event. This can be seen as a strategy to help with the trade-off between the reliability of a prediction and how quickly a prediction is acquired (criterion 5).

The third way generalisation helps learning is that should a prediction based on a generalisation be correct, then there is little need to store that additional association. This means that the data can be stored in a more compact manner, and so this helps the agent be more computationally efficient (criterion 10) by preserving memory resources.

2.4.3 Discrimination

As described above, if a stimulus is similar enough to an existing conditioned stimulus, it will also elicit the same conditioned response. This effect can be removed for particular stimuli through non-reinforcements of that similar stimulus, interspersed with reinforcements of the conditioned stimulus so that extinction does not take place for the original conditioned stimulus. Stimuli that are similar to the original stimulus in a different manner are unaffected by the discrimination training. This effect was first described by Pavlov (1927).

2.4.3.1 Analysis

The downside of generalisation is that, ultimately it is making assumptions that can easily be false. Therefore, discrimination can be seen as a process to extinguish any false assumptions. Discrimination can therefore help provide a bound to the variance allowed by generalisation (criterion 2). Discrimination also helps the agent deal with the trade-off between the reliability of a prediction and how quickly that prediction is acquired (criterion 5) in the same manner as generalisation. This phenomenon can also be seen as being a mechanism for undoing the false predictions of generalisation in the face of new evidence (criterion 4).

2.4.4 Configural Cues

When conditioning compound stimuli, the compound itself can behave as a stimulus in its own right. This can be demonstrated in a two main ways: Firstly, subjects can be conditioned such that a compound stimulus elicits the conditioned response but its component parts do not. Secondly, subjects can be conditioned such that no response is given to the compound stimulus, but any of the component parts do elicit the conditioned response. This effect was first described by Razran (1939).

2.4.4.1 Analysis

A central concept to understanding configural cues is that an agent does not know what counts as an individual event and what counts as a parallel compound event of its components. Therefore, it can be viewed that the solution

demonstrated by the configural cue experiments is to create an association with both the individual events and their corresponding compound event, inferring that compound events are events in their own right. This allows for the construction of an environmental model where the association takes place at the correct level of granularity. By compounding events in this way, the results of configural cues can be seen to be building up patterns via spatial and cross-input-type (i.e. associating a visual event with an audial event) co-occurrence (criterion 1).

The first of the two configural cue results is that component events that are learned to be not predictor events can be learned to be a predictor as a compound event. This can be seen as learning that some events are necessary but not sufficient for a prediction. From a logical viewpoint, the result can be seen as learning a conjunctive ("and") relation between the two component predictor events.

The second of the two configural cue results is that component events that are learned to be predictor events separately can be learned to not be a predictor as a compound event. From a logical viewpoint, the result can be seen as learning an exclusive disjunction ("xor") relation between the two component predictor events.

Both types of result contribute to learning a richer environmental model. However, in addition to this, the first result can also be interpreted as learning an indication that a state-dependent pattern can be predicted (criterion 8a and criterion 8b).

2.4.5 Patterning

Patterning involves intermixing presentations of a compound stimulus (i.e. one that is made up of more than one stimulus) with individual presentations of the compound stimulus' component parts. There are two forms of patterning – positive and negative. In positive patterning, a compound stimulus is presented with the unconditioned stimulus intermixed with individual presentations of the component parts that *are also* paired with the unconditioned stimulus. In negative patterning, a compound stimulus is presented with the unconditioned stimulus intermixed with individual presentations of the component parts that *are not* paired with the unconditioned stimulus. The result of positive patterning is that the compound stimulus has a stronger association strength value than the sum of its component stimuli association strength values. The result of negative patterning is that the compound stimulus has a weaker association strength value than the sum of its component stimuli association strength values. Patterning was first demonstrated by Bellingham, Gillette-Bellingham & Kehoe (1985).

2.4.5.1 Analysis

Patterning can be viewed as an extension to the results of configural cues, as they involve further interaction between compound events and their cor-

responding component events. As with the results of the configural cue experiments, the results of the patterning experiments can be interpreted as a compound event being conditioned as an independent event in its own right.

With positive patterning, where the compound event is reinforced in addition to separate reinforcements of the constituent events, leading to the compound providing a greater prediction than the sum of its parts. By assuming that a compound event is reinforced separately to its parts, this extra conditioning can be seen to be due to the separate prediction of the compound event.

With negative patterning, where the compound event is not reinforced while the constituent events are separately reinforced, leading to the compound providing a lesser prediction than the sum of its parts. By assuming that a compound event is reinforced separately to its parts, this lower conditioning can be seen to be due to the separate inhibitory prediction of the compound event.

As with the results of configural cues, a learning system that demonstrates this phenomenon would be able to learn an environmental model with a sufficient richness to be usable, and can be seen to be building up patterns via spatial and cross-input-type co-occurrence (criterion 1).

2.4.6 The Inter-Trial Interval

This phenomenon is another effect based on how the trials are timed. The inter-trial interval (I.T.I) is the time that has elapsed between reinforcements. It has been shown that increasing the length of the inter-trial interval increases conditioning. This effect was first described by Pavlov (1927).

2.4.6.1 Analysis

Longer inter-trial intervals imply that the particular event to be predicted is rarer than shorter inter-trial intervals. This in turn implies that there are fewer opportunities to learn any predictors of an event. In this case, the need to learn quickly becomes slightly more pressing than usual, due to the fewer chances to learn an association. This can be recognised by a change in the trade-off between the reliability of a prediction and the need to learn quickly (criterion 5) to be less discriminating between a true prediction and noise.

2.4.7 Facilitation of Remote Associations

Due to the drop-off of association strength as the inter-stimulus interval increases, remote associations are rarely learned. However, if another conditioned stimulus is placed in between the first conditioned stimulus and the unconditioned stimulus, to form what has been called a "serial compound", then the first (remote) conditioned stimulus can gain an association with the unconditioned stimulus. This effect was first rigorously studied by Kehoe *et al.* (1979).

2.4.7.1 Analysis

This phenomenon can be seen to be the result of an interaction between the phenomena of the inter-stimulus interval and secondary conditioning / sensory preconditioning. In this case, the result of the inter-stimulus interval effects would mean the remote predictor event would normally have a low association strength value and the closer predictor (the intermediate event) would have a larger association. Due to secondary conditioning, the intermediate event becomes associated with the remote event, meaning that the remote event becomes a predictor for the intermediate event. As the remote event is closer to the intermediate event, it is a better predictor of the intermediate event than the final predicted event. This better prediction in turn allows for a more accurate prediction of the final event because the strong association with the intermediate event demonstrates that the remote event is less likely to be a noisy relationship with the final predicted event. Therefore, through this chain of events, the predictive value of the remote event in predicting the final predicted event is larger, which is reflected in the association strength.

By allowing an increase in association strength for chains of events, a learning system can reflect the more reliable prediction of that event chain. This allows for a strategy to improve the trade-off between the reliability of a prediction and how far in advance that prediction can be made (criterion 6).

2.4.8 Primacy

In the experiment that demonstrates primacy, two conditioned stimuli are simultaneously reinforced such that both conditioned stimuli terminate at the same time, but the first stimulus is longer than the second. In this schedule, the longer first stimulus gains substantially more association strength than the shorter second stimulus. This effect was first described by Egger & Miller (1962).

2.4.8.1 Analysis

When two events predict the same event to the same reliability, but one event is longer than the other (but not so much that the penalty for long inter-stimulus intervals would cause a large difference between the two association strengths), then the longer event should be preferred as this allows for a prediction to be made further in advance without any loss in reliability (criterion 6). The primacy effect provides such a mechanism.

Within the S.B. model (Sutton & Barto, 1981), described in the next chapter, this effect has been observed to be due to a the longer event effectively receiving two reinforcements of the unconditioned stimulus for each trial: one reinforcement from the unconditioned stimulus and one reinforcement from the shorter conditioned stimulus through the effect of secondary conditioning / sensory preconditioning. This implies that the primacy effect is due to a similar mechanism as facilitation of remote associations. The S.B. model is able to

predict the primacy effect. By implementing the S.B. model as described by Sutton & Barto (1990) in a spread-sheet, where each row displays the model variables per time step, it was found that even if the predictive value of the longer event drops even slightly (for example, through introducing a short gap between the longer conditioned stimulus and the unconditioned stimulus), then the primacy effect can be observed to rapidly diminish.

2.4.9 Temporal Primacy

This is a variant of the primacy effect, and has two phases. In the first phase, a first conditioned stimulus is associated with the unconditioned stimulus. In the second phase, a second, longer conditioned stimulus is presented such that it terminates at the same time as the first stimulus. The observed result is that not only does the blocking effect fail; the introduction of the longer stimulus causes the association strength of the pre-trained stimulus to be sharply reduced. This phenomenon was first predicted by Sutton & Barto (1981) in the S.B. model, which is discussed in the next chapter, and was experimentally confirmed by Kehoe, Schreurs & Graham (1987).

2.4.9.1 Analysis

As described in the discussion of the primacy effect, primacy can be seen to be a mechanism to credit the majority of the association strength to the event that provides the longest prediction without sacrificing reliability (criterion 6). The temporal primacy effect can therefore be described as a method to transfer the majority of the association strength (and so corresponding predictive value) between an existing predictor and a new predictor that is reliably providing a longer prediction time.

Sutton & Barto (1990) noted that the prediction by the S.B. model at the time was novel and surprising, asking: "Why should a well-trained C.S. that continues to be paired with the U.S. in a good temporal relationship lose associative strength just because a new C.S. is introduced with no initial association and in a poorer temporal relationship to the U.S.?" (Sutton & Barto, 1990, pp. 508–509). Sutton & Barto did not suggest any answer to this question.

This question was briefly investigated as part of this thesis, by implementing the S.B. model within a spread-sheet where each row displays all model variables per time step. By reviewing the progress of the association strengths, it was seen that the mechanism in the S.B. model for this behaviour starts as being the same as for the primacy effect – namely that the longer stimulus increases its association strength due to a double reinforcement effect from both the unconditioned stimulus and the shorter conditioned stimulus. This accounts for the increase in the association strength of the longer stimulus.

To understand the reduction in the association strength to the shorter stimulus, note that the increase in association strength provided to the longer conditioned stimulus from the shorter conditioned stimulus is applied before the

unconditioned stimulus is presented. In the time-steps immediately prior to the presentation of the unconditioned stimulus, the sum of the two association strengths add together to form a prediction that is larger than the magnitude of the unconditioned stimulus. This leads to a negative difference between the prediction and reality that gets shared equally between the two conditioned stimuli, reducing their association strength by the same amount. The gain in association strength received by the longer stimulus is always larger than the corresponding reduction for over-prediction by the unconditioned stimulus. Therefore the net result of each trial is that the longer stimulus has an increase in association strength and the shorter stimulus has a decrease in association strength. This mechanism therefore supports the view that the purpose of the primacy and temporal primacy effects is to credit predictive capacity to the event that provides an accurate prediction for longer.

2.4.10 Learning-to-Learn

To demonstrate learning-to-learn (also known as positive transfer of learning), a two-phase experiment is used: In the first phase, a first conditioned stimulus is associated with an unconditioned stimulus. In the second phase, a second conditioned stimulus is reinforced with the same unconditioned stimulus. The rate of acquisition in the second phase will be more rapid than in the first phase. This second conditioned stimulus can be one that has been shown to not elicit any association due to the generalisation phenomenon. This phenomenon was first found in a more general sense by Harlow (1949) and then later found to apply to classical conditioning by Kehoe & Holt (1984).

2.4.10.1 Analysis

The essence of learning-to-learn in the sense used outside of conditioning is that it allows the use of previous contextual situations to speed up learning in a new context (Harlow, 1949). Using this to interpret the phenomenon within the paradigm of classical conditioning, the unconditioned stimulus can be thought of as a cue for the similarity of situation, therefore demonstrating that a novel stimulus can be a predictor of the unconditioned stimulus.

Demonstrating that an event is able to be predicted due to one predictor event increases the possibility that another predictor event exists. With each extra predictor that is observed, it increases the likelihood that a further predictor exists. Consider that if an event can be predicted by 100 predictive events, then if another candidate predictor is observed the possibility that it is a genuine predictor and not noise is subjectively more plausible than if the predicted event had only one or two known predictor events. This can be seen in the ratio between the known predictors and potential total predictors. The existence of the possible world where there are two total predictors appears more plausible than the existence of the possible world where there are 100 total predictors if one predictor has been observed ($1/2$ versus $1/100$). If 98 more predictors are then observed then it becomes increasingly more plausible

that there are 100 total predictors ($^{99}/_{100}$). Therefore, to accommodate this, a faster rate of the accumulation of association strength can be justified for increasing number of predictors. In this way, learning-to-learn can be seen to be another strategy to deal with the trade-off between the reliability of a prediction and how quickly that prediction is acquired (criterion 5).

2.4.11 Overshadowing

The salience (or intensity) of the conditioned response is also a factor in conditioning. When two conditioned stimuli are reinforced simultaneously, if one stimulus is much more salient than the other, then the conditioning only occurs with the more salient stimulus. This effect was first described by Pavlov (1927).

2.4.11.1 Analysis

Overshadowing can be seen to be a part of a strategy to deal with having more sense data that can be realistically processed (criterion 11). As described in the fourth general observation made in section 2.1, allocating the processing time for each event by the salience of each event can be a reasonable strategy to deal with having more sense data than can be processed in the time allowed. A consequence of this is that a large relative difference in the magnitude of two potential predictive events may cause the event of lower magnitude to not be processed, and so acquire less association strength than the high magnitude event.

2.4.12 Super-Conditioning

In super-conditioning, a first conditioned stimulus is reinforced in compound with a second conditioned stimulus that already has an inhibitory association with the unconditioned stimulus. The result is that the first conditioned stimulus will acquire a greater association strength value than if it were reinforced alone. This effect was first demonstrated by Rescorla (1971).

2.4.12.1 Analysis

This phenomenon can be seen as a mechanism whereby the learning system learns an exception to the prediction of a non-occurrence of an event by the presence of an inhibitory event. This contributes to the learning system by allowing a learned prediction to be undone for a specific exception (criterion 7).

By creating these exception events so that a super-conditioned event is presented alone, it produces a response greater than would be otherwise the case. This can be seen as encoding an assumption that the inhibitory event was still able to reduce the magnitude of the predicted event, even if it didn't eliminate it in the face of the strong predictor and therefore if the super-conditioned event presented alone would predict an event of greater magnitude. By encoding this assumption in the mechanism, the system is able to avoid needing

to learn to predict a greater magnitude event through experience, sacrificing some reliability for the speed that a prediction is acquired (criterion 5), whereas if the greater magnitude prediction is not the case, then presumably the over-prediction would get extinguished.

2.4.13 Backward Blocking

In backward blocking, the two phases of the blocking phenomenon are reversed. Namely, in the first phase, a two-part compound conditioned stimulus is presented with the unconditioned stimulus. In the second phase, only one part of the compound conditioned stimulus is paired with the unconditioned stimulus. The prediction is that the association strength of the part of the compound stimulus that went unpaired would diminish, due to the greater predictive capacity of the part that was subsequently reinforced. This phenomenon, contrary to some sources, was found to not occur in classical conditioning in animals (Schweitzer & Green, 1982). It was however found to occur in human causal reasoning (Chapman, 1991).

2.4.13.1 Analysis

Backward blocking can be seen as retrospectively re-evaluating the initial learned associations of the events to attribute the predictive value to the event that is observed to be a predictor outside of the pairing. This phenomenon, like blocking, would allow for the separation of noise events from predictive events (criterion 4).

2.4.14 Partial Reinforcement

Partial reinforcement experiments are where the unconditioned stimulus is paired with presentations of the conditioned stimulus on an intermittent basis. This intermittence can be either due to presenting the unconditioned stimulus randomly after the conditioned stimulus or, due to a regular pattern of non-pairing and pairing. For all ratios of reinforcements to non-reinforcements, the intensity of the conditioned response trends asymptotically with the number of trials to match the same response intensity as that of the case where all trials are reinforced. While Pavlov (1927, p. 384) did briefly mention some work hinting at the effect, it was Skinner who explored the concept in depth, looking at the effect of regular patterns of reinforcement in instrumental conditioning (Skinner, 1938).

2.4.14.1 Analysis

The results of the partial reinforcement experiment imply that associations in relation to a stochastic co-occurrence do get learned, but that this process is slower than for a deterministic association. By making the process slower and therefore requiring far more positive predictions to counter the greater number

of negative predictions, the process of acquiring a stochastic association is made robust to noise (criterion 9).

2.4.15 Partial Reinforcement Extinction Effect

When an association has been formed using a partial reinforcement schedule, it is far more resistant to extinction than one that has been formed using a full reinforcement schedule. The rate at which an association is extinguished is proportional to the probability of reinforcement – in other words, the higher the probability of reinforcement, the faster the association extinguishes. This effect was first found by Humphreys (1939) with the proportionality of the effect found by Grant & Schipper (1952).

2.4.15.1 Analysis

By allowing for the learning of stochastic relationships, it implies that those predictions that are stochastic can be allowed to have some failures without loss of association strength. By the nature of stochastic patterns, this resilience to loss of association strength needs to include some relatively long runs of failed predictions. Allowing for there to be failures without loss of association strength necessitates that extinction is harder to achieve. This allows the learning system to maintain those stochastic relationships that it has learned (criterion 9).

2.4.16 Spontaneous Recovery

When a conditioned response is extinguished and there is a long time without the presentation of the conditioned stimulus, then presentation of the conditioned stimulus occasionally elicits the conditioned response. The response is still weaker than prior to extinction, however. This effect was first described by Pavlov (1927).

2.4.16.1 Analysis

Spontaneous recovery can be seen to be a phenomenon to deal with an additional problem that arises from having to predict an association in a state-dependent manner where there may or may not be any indicators for the state of the association (criterion 8 a–d). The issue is that even if the reacquisition effect is able to adapt to a state-dependent association, if a long period of time has passed after the last extinction, it would be unknown which state is currently in play. As such, if there is a cost for not being ready for a predictive event, then in these circumstances it is prudent to partially respond to that predictive event.

2.5 Classical Conditioning Interpretations

There are two interpretations of the underlying process of classical conditioning. The two interpretations differ on what association is formed during conditioning. These interpretations are known as the stimulus-stimulus (S-S) interpretation and the stimulus-response (S-R) interpretation. The stimulus-stimulus interpretation of conditioning states that during conditioning, the concept of the conditioned stimulus becomes associated with the concept of the unconditioned stimulus, the conditioned response is then due to anticipation of the unconditioned stimulus. The stimulus-response interpretation states that the conditioned stimulus becomes directly associated with the unconditioned response. The current most widely accepted interpretation is that the vast majority of associations are stimulus-stimulus, but that in the correct conditions, stimulus-response associations can be found.

One piece of evidence in favour of stimulus-stimulus associations is that of the existence of the sensory preconditioning phenomenon. As the two neutral stimuli are associated prior to the introduction of any non-neutral stimulus, there is never any possibility for the neutral stimulus that was not paired with the non-neutral stimulus to ever gain any association with the unconditioned response, so no stimulus-response association could ever form.

Another piece of evidence against the stimulus-response interpretation is advanced by an experiment Holland & Rescorla (1975). In the experiment, first the conditioned stimulus is associated with an unconditioned stimulus of food. Once conditioning is complete, the food stimulus is devalued to the subject, either by inducing nausea from spinning the subject around, or by allowing the subject to feed until it did not want any more food. The result was that the conditioned response was no longer elicited by the conditioned stimulus. If the stimulus-response interpretation was correct, then the lack of value of food would not stop the response from occurring.

A third piece of evidence is provided by Siegel (1975), whose study lends support to a proposition that the unconditioned response is determined to be the response that compensates for the effects of the unconditioned stimulus. The association of the conditioned stimulus then allows for that compensatory response to be initiated prior to the arrival of the unconditioned stimulus. Siegel gave rats repeated injections of insulin, each reducing the blood-glucose level. In this experiment, the act of injecting was the conditioned stimulus and the reduction of blood-glucose was the unconditioned stimulus. To test the association, the rats were given a saline injection and the blood-glucose level was monitored and found to increase. This showed that the conditioned response was compensating for the unconditioned stimulus, rather than the same as the unconditioned stimulus. This lends further evidential weight against the stimulus-response interpretation (and in absence of a third theory, in favour of the stimulus-stimulus interpretation), as if the formed association was a direct association between the conditioned stimulus and the unconditioned response, the conditioned response would need to be the same.

The condition in which stimulus-response associations have been observed is in second-order conditioning experiments. This is firstly shown in a study by Rescorla (1973). Rescorla's study used the concept of habituation to study the interpretations. Habituation is where the response elicited by a stimulus is weakened through repeated exposure to the stimulus. It should be noted that this phenomenon is not in itself an effect associated with conditioning, as it does not affect the learned association between stimuli. In Rescorla's study, a CS-US association was reinforced, and then using habituation, the unconditioned stimulus was weakened. The conditioned response was also found to be weakened. This substantiated the stimulus-stimulus interpretation. However, when the experiment was repeated with a different group of subjects using a phase of second-order conditioning with the first conditioned stimulus, and then weakening the unconditioned response through habituation, the conditioned response of the second conditioned stimulus is not weakened, substantiating the stimulus-response interpretation.

A second experiment demonstrating a stimulus-response association in second-order conditioning was conducted by Holland & Rescorla (1975). In applying the same stimulus devaluation of food as they had previously done in first-order conditioning to second-order conditioning, the conditioned response of the second order stimulus was demonstrated to not weaken under unconditioned stimulus devaluation.

Not all second-order associations are stimulus-response associations however. In an experiment by Rashotte, Griffin & Sisk (1977), a second order conditioned stimulus was created, and then the original first-order association was extinguished. This extinction led to a considerable reduction in the conditioned response of the second-order conditioned stimulus. The stimulus-response interpretation is disconfirmed in this experiment as extinction of one association should not have led to the extinction of another, independent association. This apparently contradictory evidence regarding second-order associations was attempted to be reconciled by Holland (1985) by arguing that the stimulus and response aspects of stimuli compete for association, the winner determined by the nature of the experiment.

2.6 Chapter Conclusion

This chapter has reviewed and analysed the background psychology that is used within this thesis. The ideas presented in this chapter are used in both chapter three and chapter four. Chapter three presents the work more directly related to this thesis. As part of that presentation, existing models of classical conditioning from the psychology community are reviewed, which require the background knowledge of classical conditioning presented in this chapter to understand. The ideas in this chapter inform the overall design of the system presented in chapter four and due to the system directly implementing some of the phenomena of classical conditioning.

Through an analysis of the features a passive learning system needs to function, a set of criteria have been established. These criteria have then been used in an analysis of classical conditioning, the analysis arguing that the phenomena of classical conditioning can be seen to satisfy those criteria. Table 2.2 summarises both analyses by tabulating the classical conditioning phenomena and which criteria each phenomenon contributes to. Every criterion in the table is contributed to by at least one phenomenon of classical conditioning. As stated in the introduction, the purpose of this analysis is to provide both a thorough grounding for the hypotheses and the basis for which the ideas will be used within the system presented in chapter four.

On the not-fully-settled debate of the two interpretations of classical conditioning, this thesis sides with the most widely accepted interpretation that while both stimulus-stimulus and stimulus-response associations can exist, the majority of associations are stimulus-stimulus in nature. However, the ideas presented in this thesis are based upon the stimulus-stimulus interpretation for two reasons. The first reason is due to the widely-held view that the majority of associations are of the stimulus-stimulus form. The second reason is that this interpretation was believed to be the most compatible with developing a passive learning system. Therefore, the analysis of the phenomena presented in this chapter comes from that viewpoint, which also means the system presented in chapter four follows from the same viewpoint.

This chapter presented and analysed the background knowledge that is used, but that the project that this thesis reports on did not seek to contribute towards. The next chapter looks at the work done in the areas that the project this thesis reports on did seek to contribute knowledge to.

		Criterion														
		1	2	3	4	5	6	7	8a	8b	8c	8d	9	10	11	12
Phenomenon	Acquisition	✓			✓									✓		
	Extinction				✓			✓								
	The Inter-Stimulus Interval						✓	✓						✓	✓	✓
	Reacquisition								✓	✓	✓	✓				
	Blocking				✓											
	Recovery from Blocking				✓											
	Conditioned Inhibition								✓		✓					
	Extinction of Conditioned Inhibition				✓			✓								
	Latent Inhibition				✓											
	U.S.-Pre-Exposure Effect				✓											
	Sensory Preconditioning			✓			✓									
	Secondary Conditioning			✓			✓									✓
	Contingency				✓			✓								
	Generalisation		✓			✓								✓		
	Discrimination		✓		✓	✓										
	Configural Cues	✓							✓	✓						
	Patterning	✓														
	The Inter-Trial Interval					✓										
	Facilitation of Remote Associations						✓									
	Primacy						✓									
	Temporal Primacy						✓									
	Learning-to-Learn					✓										
	Overshadowing														✓	
	Super-Conditioning					✓		✓								
	Backward Blocking				✓											
	Partial Reinforcement												✓			
	Partial Reinforcement Extinction Effect												✓			
	Spontaneous Recovery								✓	✓	✓	✓				

Table 2.2: A matrix summarising which phenomena of classical conditioning contribute to each of the criteria.

Chapter 3

Related Work

This chapter will review the ideas related to this thesis. This chapter begins by looking at the area of classical conditioning that is most related to this thesis, that of the theoretical models of classical conditioning. These models try to explain the diverse range of phenomena within classical conditioning. The system presented in chapter four does not attempt to create any viable, or even a unified explanation, of classical conditioning. However, it does in some sense model classical conditioning and therefore this chapter will review these models.

Stated in both hypotheses is that the aim is to learn commonsense knowledge. Therefore this chapter will next discuss commonsense knowledge, including the existing methods of representing it that are relevant to this thesis and the existing attempts to acquire it.

There already exists methods within artificial intelligence that use conditioning as the basis for learning. These methods are collectively known as reinforcement learning methods. As this thesis and those existing methods both derive from a common ancestor, the third section will review these existing methods.

In order for the system presented in chapter four to function it needs input from the environment that it is observing. The form of input that has been selected is that of tracked object data. There have been studies within the field of computer vision that have also looked into learning to predict events formed from visual data. A review of these studies is briefly discussed in the last section of this chapter, along with a brief discussion of the types of tracking methods that exist.

3.1 Models of Classical Conditioning

Ever since the start of research into classical conditioning, models have been proposed that attempt to create a unified explanation for the diverse range of phenomena of classical conditioning.

One approach in categorising these models is to look at when the change in association strength between the stimuli is computed. In trial-level models, the computation is dealt with after all relevant stimuli have terminated. In real-time models, the computation happens at every time-frame, and can cope with those frames being arbitrarily small.

This section covers the most influential models that have been proposed. These shall be covered in approximately chronological order. The models that are described are: The Stimulus Substitution Model, The Rescorla-Wagner Model, Mackintosh's Attentional Model, The Sometimes-Opponent Process, The Sutton-Barto Model and The Temporal Difference Model. The section will finish with an overview of several neural network-based models.

Alonso & Schmajuk (2012) have proposed a standardised list of classical conditioning phenomena with the aim of standardising how computational models of classical conditioning are tested. In a special issue of Learning and Behaviour, focusing on computational models of classical conditioning, a number of the models mentioned below are tested against this standardised list. This includes, most notably for this thesis, the TD model of classical conditioning (Ludvig *et al.*, 2012).

3.1.1 The Stimulus Substitution Model

The first explanation of classical conditioning was developed by Pavlov (1927), and has since been known as the stimulus-substitution model, a trial-level model. This model proposes that during conditioning, the brain areas representing the conditioned stimulus and those representing the unconditioned stimulus develop a connection. This connection ensures that if the area of the brain representing the conditioned stimulus is stimulated then the area of the brain responsible for the unconditioned stimulus will also be stimulated. In turn, as the unconditioned stimulus area is stimulated then the area that causes the unconditioned response will be stimulated. Pavlov effectively assumed that the conditioned response is simply a manifestation of the unconditioned response.

This model was later discredited as unconditioned stimuli were found where the unconditioned response is different to the conditioned response. An example of such an unconditioned stimulus is an electric shock applied to a rat. When a rat is presented with an electric shock its unconditioned response is to attempt to flee. The conditioned response however, is to freeze. While this model is discredited, its ideas have influenced later theories and models.

3.1.2 Rescorla-Wagner Model

The most influential model is known as the Rescorla-Wagner model (Rescorla & Wagner, 1972). This trial-level model is both simple and yet able to account for a very wide range of the phenomena. The basis of the model is a single function, shown in equation 3.1.

$$\Delta V = \alpha\beta\,(\lambda - V) \tag{3.1}$$

In the equation, V is the association strength between the conditioned stimulus and the unconditioned stimulus – i.e. the amount of conditioning the subject displays; ΔV is the change in the association strength; α is the rate of learning for the conditioned stimulus (with the constraint $0 < \alpha < 1$); β is the rate of learning for the unconditioned stimulus (with the constraint $0 < \beta < 1$) and λ is the maximum possible association strength for the unconditioned stimulus.

The formula is applied iteratively. A reinforcement of the conditioned stimulus with the unconditioned stimulus is represented as an iteration of the formula. This leads to the association strength asymptotically approaching the maximum – similar to the sigmoidal shape of acquisition. To model extinction, when the unconditioned stimulus is not present, the maximum possible association strength (λ) will be zero. By the equation, if the total association strength (V) is greater than zero due to previous trials, then the overall change in the association strength will be negative. This negative ΔV will cause a reduction in association strength, as shown in extinction experiments.

The model also covers the case where there is more than one neutral stimulus involved. Where there is a compound stimulus, the basic equation is adapted as shown in equations 3.2 to 3.4.

$$V_{AB} = V_A + V_B \tag{3.2}$$

$$\Delta V_A = \alpha_A\beta\,(\lambda - V_{AB}) \tag{3.3}$$

$$\Delta V_B = \alpha_B\beta\,(\lambda - V_{AB}) \tag{3.4}$$

In this set of equations, V_A corresponds to the association strength of the first conditioned stimulus; ΔV_A is the change in the association strength of the first conditioned stimulus; V_B is the association strength of the second conditioned stimulus; ΔV_B corresponds to the change in the association strength of the second conditioned stimulus; V_{AB} corresponds to the association strength of the compound stimulus; α_A is the learning rate for the first stimulus; α_B is the learning rate for the second stimulus; β and λ are as before.

The different parts of the compound stimulus effectively compete to be associated with the unconditioned stimulus. If one of the two stimuli is reinforced more than another, then that stimulus gains more of the total association strength available. Similarly, if one stimulus is conditioned before it is used within a compound stimulus, the other stimulus of the compound stim-

ulus will gain little association strength. This is consistent with the blocking phenomenon.

The compound stimuli version of the model can predict a number of the other phenomena related to compound stimuli. For instance, overshadowing can be created by varying the learning rate of each conditioned stimulus in proportion to the salience of the stimulus. The lower learning rate of two stimuli in the compound stimulus will gain association strength slower than the other. Another example is conditioned inhibition. First a strong association V_A is formed with a stimulus, and then that stimulus is non-reinforced with a second stimulus. This means that V_A is high, but $V_{AB} = 0$. As $V_{AB} = V_A + V_B$, V_B must be negative. This negative association strength allows the model to predict conditioned inhibition.

As with all models, there are some phenomena that elude the Rescorla-Wagner model. The overview by Miller *et al.* (1995) gives a comprehensive review of the successes and failures of the Rescorla-Wagner model. Most notably the phenomena associated with pre-exposure. Both latent inhibition and the U.S.-pre-exposure effect cannot be predicted. This is because the model only deals with association strengths which are assumed to start at zero. Secondary conditioning is also not predicted. This is because the model predicts that where a stimulus occurs, whether prior or simultaneously, with the conditioned stimulus but without the unconditioned stimulus, that stimulus should become inhibitory.

Despite its shortcomings, the Rescorla-Wagner model continues to influence both theoretical and experimental approaches within the study of classical conditioning. Alonso *et al.* (2012) produced a computer simulation of a version of the Rescorla-Wagner model that includes an extension that allows for configural cues to be modelled.

3.1.3 Mackintosh's Attentional Model

Mackintosh (1975), argued that instead of stimuli competing for associability with the unconditioned stimulus, relevant stimuli instead gain attention of the subject in accordance with their relevance. The subject would give a response determined by only those stimuli that are attended to. It is then the attention level for each stimulus that would change with how much that stimulus preceded an unexpected unconditioned stimulus event.

While the idea that the attention the subject gives a stimulus is related to its relevance was not new, Mackintosh noted a discrepancy between an assumption made by a number of those previous models and the experimental data that had been produced up until that point. The assumption is that there is a competition between stimuli for attention i.e. the increase in attention for one stimulus reduces the attention given to other stimuli. Mackintosh also argued that this criticism applied to the Rescorla-Wagner model because, while there is no use of attention in the Rescorla-Wagner model, stimuli still compete for a limited resource – the pool of associative strength.

Mackintosh then proceeded to propose a trial-level model that can explain blocking and overshadowing without the use of a limited resource – as an attention-based extension to the Rescorla-Wagner model. Firstly, only the basic rule is used for all stimuli, with no special rule for compound stimuli – though the learning rates are still stimulus-specific. Secondly, part of the conditioned-stimulus-specific learning rate would be related to the attention the subject gives that stimulus in the specific trial. Lastly, another part of the conditioned-stimulus-specific learning rate would be increased if that stimulus is the best available predictor of the unconditioned stimulus and decreased if it is no better than any of the other stimuli. This last point is codified in equations 3.5 and 3.6.

$$|\lambda - V_A| > |\lambda - V_X| \Rightarrow \Delta\alpha_A > 0 \tag{3.5}$$

$$|\lambda - V_A| \leq |\lambda - V_X| \Rightarrow \Delta\alpha_A < 0 \tag{3.6}$$

In these rules, $\Delta\alpha_A$ is the change in the learning rate for stimulus A; V_A is the association strength of stimulus A; V_X is the association strength of all other stimuli other than A present in the reinforcement trial; and λ is the maximum possible association strength. Mackintosh also suggested that the size of the change would be proportional to the magnitude of the difference between $|\lambda - V_A|$ and $|\lambda - V_X|$.

In this model blocking can still be achieved as, if a strong association is established, then any other stimuli that subsequently follow would quickly be stopped from gaining any association strength as the learning rate would be at or close to nothing. Overshadowing can still be achieved as salience is also a factor included in the learning rate. If a stimulus is salient enough, its starting learning rate will be faster than other stimuli. A faster learning rate will be able to force the learning rates of other stimuli down to be at or close to zero.

Mackintosh argued that this model is also able to account for pre-exposure effects such as latent inhibition, phenomena that the Rescorla-Wagner model is not able to predict. The U.S.-pre-exposure effect is predicted through the introduction of a "background" or "environmental" stimulus. When the unconditioned stimulus is reinforced alone, it is instead reinforced with the background stimulus. This allows the background stimulus, in effect, to act as a blocking stimulus for future C.S.-U.S. pairings.

Latent Inhibition is due to the second equation that governs part of the learning rate. This is because the learning rate declines even when a stimulus is equal in association strength to another stimulus. If a conditioned stimulus is presented without any sort of reinforcement prior to any pairing with an unconditioned stimulus, then both the maximum association strength and current association strength would be zero for all stimuli. As both sides of the inequality would be zero for all stimuli, the corresponding learning rate for each stimulus would reduce.

A weakness of this model was demonstrated by an experiment by Hall & Pearce (1979). The model predicts that the best available predictor of reinforcement will receive the greatest increase in attention after reinforcement, allowing for the best predictor to be learned the fastest. Hall & Pearce (1979) conducted an experiment of two phases. In the first phase, a first conditioned stimulus was paired with an unconditioned stimulus of a weak electric shock. In the second phase, the subjects were split into two groups: The first group had the same conditioned stimulus as the first phase paired with an unconditioned stimulus of a strong electric shock. The second group had a novel conditioned stimulus paired with an unconditioned stimulus of a strong electric shock. If the attention model was correct, the group provided with the first conditioned stimulus would learn the association with the strong electric shock faster than the group with the novel stimulus, as the first conditioned stimulus is already the best available predictor of an electric shock. The results of the experiment showed that the converse is the case: the group exposed to the novel conditioned stimulus learned faster. The reason the two groups did not learn at the same rate was explained to be due to the first phase effectively acting as latent inhibition trials for the first group.

3.1.4 The Sometimes-Opponent-Process

The sometimes-opponent-process (sometimes referred to as the S.O.P. model) was a real-time model proposed by Wagner (1981; Donegan & Wagner 1987) as an extension of the opponent process theory. The opponent process theory itself is not directly related to classical conditioning, but instead is a theory of the response to unconditioned stimuli. In order to explain the sometimes-opponent-process, the opponent process theory shall be described first.

The opponent process theory was initially introduced by Epstein (1967) and heavily expanded upon by Solomon & Corbit (1974). The theory proposed that, for stimuli that produce a reaction, the subject produces two separate reactions that oppose each other. If the first reaction is pleasant, then the second will counteract that by being unpleasant and visa-versa. The first reaction, the A-state, is high intensity; onset is immediate and diminishes rapidly after the stimulus event has passed. The second reaction, the B-state, is low intensity; onset is delayed and diminishes slowly after the stimulus event has passed. The resultant effect on the observed response is that the initial response to stimulation is high intensity (the B-state has yet to begin), it then comes down to a plateaux while the stimulus event is in effect (the B-state counteracting the A-state), and then the response rapidly reverses and slowly declines after the stimulus event has passed (the decay of the B-state being longer than the A-state).

As the subject receives many stimulations, while the A-state does not change, the B-state increases in intensity and onset becomes more immediate (though it still diminishes slowly after the stimulus event has passed). The resultant effect on the observed response after many stimulations, is that the

overall response during stimulation is lower in intensity (the B-state beginning earlier and is more intense so better at counteracting the A-state), with an overall opposite response after stimulation that is intense and prolonged (the B-state being more intense but with the same decay). Solomon & Corbit (1974) suggested that the B-state is a process that allows the subject to avoid extreme states of arousal, as they are resource-intensive to sustain.

Epstein (1967) used sky-diving as an example of the opponent process at work. When a novice skydiver jumps, they feel terror. After the jump is successful, they feel a pleasurable feeling of relief. A second example of the opponent process at work is that of recreational drug use such as alcohol. The initial, A-state response is pleasurable, whereas the B-state works to counteract that and manifests itself in withdrawal symptoms after the drug's effects have ended.

Wagner's sometimes-opponent process model (Wagner, 1981), while starting from a different perspective, was later reinterpreted as an extension of opponent process theory (Donegan & Wagner, 1987). It was this re-interpretation that became influential. Wagner's primary motive was to explain why the conditioned response is sometimes different to the unconditioned response. Wagner argued that all representations of stimuli are collections of a large number of elements that each can be in one of three states: Inactive (I), a first active state (A1) and a secondary active state (A2).

When a stimulus representation is activated by the direct observation of that stimulus, the majority of the elements enter their A1 state and slowly decay first to the A2 state and then back to the inactive state. When a stimulus representation has the majority of its elements in either A1 or A2 state, then it induces any other stimulus representations that it is associated with that are mainly in their inactive state to be mainly in their A2 state. This inducement occurs in proportion to the association strength between the two representations. When two representations are mainly in their A1 state, then the association strength between the two representations increases.

The manifest responses when the majority of these elements are in the A1 and A2 states are argued to correspond to the A-state and B-state of the opponent process. As with the opponent process, both active states of the representation of an unconditioned stimulus have their own set of responses. Unlike the opponent process however, these responses can either be the same or opposed, dependant on the nature of the unconditioned stimulus. As representations of stimuli can only ever set another representation to be mostly in its A2 state, when the representation of a conditioned stimulus induces that of an unconditioned stimulus, then only the A2 response is elicited.

The sometimes-opponent-process allowed for the prediction of the timing effects of classical conditioning, namely the inter-stimulus interval and the inter-trial interval. The efficacy curve of the inter-stimulus interval is produced by this model by the fact that association strength is only gained when the majority of the elements of both stimulus representations are in their A1 state

and that it takes time for the majority state to change. The upward curve at the beginning of the inter-stimulus interval caused due to one or both elements transitioning from inactive to their A1 state, the downward curve then caused by the slow decay from A1 to A2.

There is a similar explanation for the inter-trial interval. When elements of a representation have yet to fully decay back to their inactive state, then there are fewer elements available to transition from the inactive state to the A1 state. The result of this is that for short inter-trial intervals, the A1 state has fewer elements than would otherwise be for a longer inter-trial interval, and so the rate of increase of the association strength is lower. As the inter-trial interval increases, there are more inactive elements available to transition to the A1 state and thus a greater rate of increase of the association strength.

Once strong piece of evidence for the sometimes-opponent-process came from Thompson (1986) who showed that with the eyeblink response in rabbits (the stimulus being a puff of air to the eye), there are in fact two separate components to the blink, one with latency of 20ms and one with latency of 70ms. This agrees with Wagner's proposal that even when there is no opponent response, there are still two components that just happen to be the same response.

A criticism of the sometimes-opponent-process, later reported by Wagner & Brandon (1989), is that it predicts that all responses to a particular unconditioned stimulus will each peak at the same inter-stimulus interval time. Vandercar & Schneiderman (1967) reported that when a rabbit is given a puff of air to the eye, the conditioned response of the eyeblink peaks at a different inter-stimulus interval time than to the conditioned response of an increased heart rate. Wagner & Brandon (1989) proposed an extension to the model, known as A.E.S.O.P. to account for these issues. That extension makes use of an emotional response and so is out of scope for this section.

3.1.5 The Temporal Difference (T.D.) Model and the Sutton-Barto (S.B.) Model

As the field of conditioning models progressed, it began to be influenced by ideas from within the artificial intelligence community. This is apparent in the Temporal Difference model, a real-time model, which was presented by Sutton & Barto (1987; 1990). The model was an application of the T.D. method of reinforcement learning, a method from artificial intelligence developed by Sutton (1984; 1988) as a method of assigning credit for a reward or punishment to prior actions taken by an agent. This method is extensively reviewed in later in this chapter.

The T.D. method of machine learning was itself influenced by and developed from an earlier real-time model of classical conditioning by Sutton & Barto that became known as the S.B. model (Sutton & Barto, 1981), a model based on the ideas of Klopf (1972). A short overview of Klopf's work forms part of section 3.1.6.

In (Sutton & Barto, 1990), the S.B. model was described in a different manner to the original paper, which provides a clearer way of understanding the model's operation. The new description placed the S.B. model in the context of an observation that was made about a large set of models of classical conditioning. The observation is that many models of classical conditioning have the functional form shown in equation 3.7.

$$\Delta V = \text{Reinforcement} \times \text{Eligibility} \tag{3.7}$$

As usual, ΔV represents the change in association strength. "Reinforcement" is defined loosely as the level of unconditioned stimulus processing. "Eligibility" on the same lines was loosely defined as the level of conditioned stimulus processing. Sutton & Barto argued that many models focus primarily at one or the other part, but rarely both. For example, the Rescorla-Wagner model can be said that the "α" part of the formula corresponds to eligibility and the $\beta\left(\lambda - V_{AB}\right)$ part of the formula corresponds to reinforcement. In the Rescorla-Wagner model, the model can be argued to look primarily to the reinforcement side of the function. An example of a model that primarily deals with eligibility would be Mackintosh's attention model.

In the S.B. model, both parts of the function were used extensively. The reinforcement part used an equation Sutton & Barto later named the $\dot{Y}$ theory (pronounced "Y dot") (Sutton & Barto, 1990). In the $\dot{Y}$ theory, every stimulus S produces a reinforcement of $+V_S$ on onset, $-V_S$ on offset and zero at all other times. The value V_S represents the association strength value of the stimulus. The unconditioned stimulus has a fixed, positive association strength value with itself; all other stimuli have a starting association strength value of zero. Time is assumed to pass in small increments. The function $\dot{Y}\left(t\right)$ is defined to be the sum of all reinforcement values that have occurred at time t. The resultant value of $\dot{Y}\left(t\right)$ is then used as the reinforcement part of the ΔV equation.

For the eligibility part of the S.B. model, Sutton & Barto used the concept of an eligibility trace that was first developed by Klopf (1972). An eligibility trace is a time-dependant function that describes the eligibility of a given stimulus in relation to the timing of the presentation of that stimulus. The eligibility trace used by the S.B. model builds while the stimulus is present and then decays when it is removed. In order to do this, the S.B. model represents the presence and non-presence of a conditioned stimulus S in terms of a variable $X_S\left(t\right)$, which is defined at time t to be one when the stimulus is present and zero otherwise. The eligibility trace $\overline{X_S}\left(t\right)$ at time t is then defined as a running average of the values of $X_S\left(t\right)$, as shown in equation 3.8.

$$\overline{X_S}\left(t-1\right) = \overline{X_S}\left(t\right) + \delta\left(X_S\left(t\right) - \overline{X_S}\left(t\right)\right) \tag{3.8}$$

Where δ is defined to be the weighting placed between the present value of $X_S\left(t\right)$ and past values. The S.B. model then puts these two components,

the $\dot{Y}$ theory and the eligibility trace together to form a single update equation to be applied for each stimulus S at time t, as shown in equation 3.9.

$$\Delta V_S = \beta \dot{Y} \times \alpha_S \overline{X_S} \tag{3.9}$$

The successes of the S.B. model was that it did predict all of the Rescorla-Wagner model's phenomena, plus was able to deal with inter-stimulus-interval effects, and predicted the existence of the temporal primacy effect, which was subsequently confirmed experimentally. However, there arose two major problems with the model. Firstly, when the I.S.I. is very short and the stimulus durations were short (i.e. only a few time-steps) and overlapped, the association gained becomes inhibitory. This prediction was disconfirmed experimentally prior to the model being published. It was not found for some time as only stimuli that were active for much longer time-steps were tested.

The second problem arises in trials where the conditioned stimulus continues for a variable length of time but the unconditioned stimulus always starts as the conditioned stimulus stops. The observed experimental effect is that the association strength between the two stimuli reduces as the duration of the conditioned stimulus increases. The prediction by the S.B. model however, is that the duration of the conditioned stimulus does not affect the strength of association in this type of trial.

There were a number of attempts to rectify both of these problems with the model. These attempts were described by Sutton & Barto (1990). However, none of the modifications of the theory were completely satisfactory. With this in mind, Sutton & Barto proposed a model that while sharing some similarities with the S.B. model, has a very different basis. This model was known as the Temporal Difference (T.D.) model of conditioning.

As described before, the T.D. model is a solution to assigning credit of present awards to the correct past actions. The T.D. model does this by attempting to predict an imminence-weighted sum of all future unconditioned stimuli.

When attempting to predict future unconditioned stimuli, ideally, one would wish to predict all future unconditioned stimuli so as to apply those to the current actions or stimuli; however this becomes increasingly difficult as the prediction goes further into the future. Therefore, at a given time-step, the prediction should be weighted more towards the next time-step, then slightly less for the time-step after that and so on. This means that as an unconditioned stimulus becomes more imminent, the prediction that it will happen at the next time step should be greater. This also means that for unconditioned stimuli that last more than one time-step, when the current time-step is in the middle of an unconditioned stimulus, the strength of prediction should be in rough accordance with how many future time-steps remain of the stimulus.

Algebraically, Sutton & Barto (1990) expressed this in formula shown in equation 3.10.

$$\overline{V_t} = \lambda_{t+1} + \gamma\lambda_{t+2} + \gamma^2\lambda_{t+3} + \gamma^3\lambda_{t+4} + \cdots \tag{3.10}$$

Where $\overline{V_t}$ is the prediction made at time t, λ_t denotes the level of intensity of the unconditioned stimulus and γ is the imminence weighting, $0 \leq \gamma < 1$, with which smaller values denoting a greater weighting to immediate values. Through algebraic manipulation, this can be written in the form shown in equation 3.11.

$$\overline{V_t} = \lambda_{t+1} + \gamma\overline{V_{t+1}} \tag{3.11}$$

This formula denotes the ideal level of prediction at any given time step. Therefore, the discrepancy between the current prediction and what it should ideally be is the level of reinforcement that should be provided on any particular time-step, as shown in equation 3.12.

$$\text{Reinforcement} = \lambda_{t+1} + \gamma\overline{V_{t+1}} - \overline{V_t} \tag{3.12}$$

This can then be used instead of the $\dot{Y}$ theory of the S.B. model to provide the association strength update formula for the T.D. model, as shown in equation 3.13.

$$\Delta V_i = \beta\left(\lambda_{t+1} + \gamma\overline{V_{t+1}} - \overline{V_t}\right) \times \alpha_i\overline{X_i} \tag{3.13}$$

Sutton & Barto showed that this model is able to predict all the same phenomena of the S.B. model without the problems that were encountered with the S.B. model. However, the model is not able to predict several classes of phenomena – many of which have been discussed by Sutton & Barto (1990). The phenomena that were discussed include configural cues, overshadowing and sensory preconditioning.

It is also believed that the T.D. model would not be able to predict the pre-exposure effects of latent inhibition and the U.S. pre-exposure effect. The reasoning for this claim is that there is no state in the model that can record the non-pairings of stimuli that would allow for pre-exposure effects to be included.

3.1.6 Artificial Neural Network Models

Of the ideas within artificial intelligence, it is that of artificial neural networks that has had the most influence over models of classical conditioning. However, the models that use artificial neural networks are out of the scope of this thesis. This is because it would be circular to argue that the phenomena that make up classical conditioning can be used for machine learning in a manner that is independent of implementation and then test this idea by implementing those phenomena by using a well-established method of machine learning. While this argument could also apply to the T.D. model, the ideas presented in this thesis are sufficiently close to warrant a discussion of the differences between

that model and those used in this thesis – this is not the case for artificial neural networks. As they are out of scope, but make up a large body of the work on models of conditioning, these models shall only be covered by a short overview.

The earliest work on modelling classical conditioning as an artificial neural network came from Grossberg (1969; 1974). Grossberg's early work attempted to derive a real-time general model of learning by applying well-defined constraints or postulates about how the neurons and the network they compose must act, each based either on observation or deductive argument. The predictions of the model were then compared with psychological and neuroscientific phenomena, including conditioning. Later on, Grossberg proposed the concept of a gated dipole – a sub-structure of an artificial neural network whereby the onset of an event and its offset compete to condition to various stimuli signals and drive signals that are active at the same time. When this is used as part of a larger network, the network can learn to signal the expectation of the imminent presentation of a given stimuli based on present stimuli. An overview of this work can be found in (Grossberg, 1982). With Carpenter, Grossberg developed a highly successful artificial neural network theory known as Adaptive Resonance Theory (ART) (Carpenter & Grossberg, 1987). This theory and accompanying neural networks are artificial neural networks that address categorisation and prediction problems.

Starting at a similar time, but independently of Grossberg, Klopf (1972) described a real-time neuronal model based on cybernetic principles. Klopf proposed that a network that is composed of components that seek to maximise some metric[1] will itself as a whole network seek to maximise a metric (that could be the same or different to that of the component) . From such a model, Klopf argued that the phenomena of classical and instrumental conditioning arise as epiphenomena, along with the phenomenon of habituation. Later, Klopf was influenced by the S.B. model (Sutton & Barto, 1981) (which in turn was influenced by Klopf's earlier work) to create another model of conditioning, which Klopf named the "drive-reinforcement model" (Klopf, 1988). In the drive-reinforcement model, Klopf mixed a variant of the S.B. model with the ideas from the Hebbian neuronal model (Hebb, 1949). In doing so, Klopf changed the Hebbian neuronal model, primarily by having the neuron correlate the derivative of delayed pre- and post-synaptic activity levels, rather than directly correlating immediate pre- and post-synaptic activity levels.

A large contribution to the field of neural-network-based models of conditioning has been made by Schmajuk. Schmajuk's first contribution to the field was in conjunction with Grossberg (Grossberg & Schmajuk, 1987). This

[1]Klopf termed these metric-maximising components "heterostatic". This phrase appears to be used to differentiate from the theory of homeostasis – where a component seeks to maintain a given state. However this name is too broad for how it is used – a chaotic system that neither maintains nor maximises its state could also be called "heterostatic".

model expanded Grossberg's gated dipole model, adding an associative learning mechanism. This model was then used to predict phenomena such as blocking and overshadowing in a real-time manner.

Of the models produced by Schmajuk, the most notable are the G.S. model, the S.D. model, and the S.L.G. model. The G.S. model by Grossberg & Schmajuk (1989) is again, an augmentation of Grossberg's earlier work, though is not an expansion of the previous collaboration between the two. The G.S. model relies on its real-time nature by adding an array of neurons that each peak at slightly different times, giving a spectrum of peaks that can then be associated with stimuli and drive signals to give timing predictions between stimuli. This model predicts both the drop in association for stimulus pairings paired with very short inter-stimulus intervals and the drop in association for stimulus pairings paired with long inter-stimulus intervals.

The S.D. model by Schmajuk & DiCarlo (1992) uses a real-time biologically plausible version of the standard three-layer, artificial neural network that uses the back-propagation algorithm. The most notable differences are firstly, the input units connect both to the hidden layer and a single output layer unit directly. Secondly all output layer units only receive input from a single lower-layer unit. Thirdly, the back-propagation is implemented in real-time as an external set of recurrent units that compare the outputs from the output layer with the expected output and uses the error difference to update the weights of the output and hidden layers. Each layer of the model was then mapped to various regions of the brain and simulations of lesions to each layer were made. The model was found to match both lesion data and able to predict a number of the phenomena of classical conditioning, including patterning and generalisation.

The S.L.G. model by Schmajuk, Lam & Gray (1996) was designed in order to model in real-time the data relating to the phenomenon of latent inhibition. The model works due to the feedback between several different networks. An attentional system controls how fast the model of the environment adapts. The model of the environment attempts to predict future presentations of stimuli. The error between the expectations of the model and what actually happens feeds into a novelty system. The total novelty at that particular time then feeds into the attentional system. By using total novelty, the system can predict latent inhibition because stimuli that have been encountered earlier have less novelty than those that have not. A lower total novelty feeds into the attentional system and so changes to the model are slower. Schmajuk's more recent work develops variants of this model (Schmajuk, 2005; Schmajuk *et al.*, 2010; Kutlu & Schmajuk, 2012).

There are a couple of other notable neural network models that are by authors that do not have a wider corpus of work on models of conditioning. The first of these is by Pearce & Hall (1980). Pearce & Hall contended that unlike the Mackintosh (1975) model, the rate of association is related to the reliability of a conditioned stimulus to predict its own consequences. Pearce &

Hall represented this as a network of information flows that updated in a trial-level manner. The other notable model, a trial-level model, is by Kehoe (1988). Kehoe's model is probably the first to test the standard three-layer artificial neural network against classical conditioning phenomena.

3.2 Commonsense Knowledge

For an autonomous agent to interact rationally within its environment, it must have access to usable knowledge regarding that environment. Commonsense knowledge is that knowledge of the world that appears to be self-evident to humans, such as "Trees are usually green" and "Things that go up, later come down". The concept of commonsense knowledge was first discussed by McCarthy (1959), defining it as follows:

> *"A program has common sense if it automatically deduces for itself a sufficiently wide class of immediate consequences of anything it is told and what it already knows."*
>
> *(McCarthy, 1959, p. 78)*

This definition points out two central problems in the field of commonsense knowledge. The first problem is that commonsense knowledge needs to be encoded such that any piece of knowledge is easily accessible to the computer. The form in which the knowledge is encoded places a limitation on how quickly any one piece of knowledge can be used in any automatic deduction. As an analogy, the way that a piece of knowledge is encoded in the mind will not bear any resemblance to that knowledge written down, and in the mind a piece of knowledge (a knowledge rule) can usually be accessed far faster than accessing it in the written form.

The second problem is that it is not sufficient for that knowledge to be simply encoded in a machine readable manner; a system that has commonsense knowledge also needs to be able to use it. A program with commonsense knowledge needs a mechanism to make deductions and inferences that can be made from the knowledge encoded within that program. These two problems typically need to be tackled simultaneously as decisions made in tackling one problem will affect the choices available in the second. The first problem is known as representation and the second is known as reasoning.

McCarthy (1959) proposed that the general solution to the twin problems of representation and reasoning is the use of predicate logic, though the inference system he proposed was incomplete. The inference system McCarthy proposed has been greatly improved over the years, with the resolution rule (Robinson, 1965), Prolog (Colmerauer *et al.*, 1973; Colmerauer & Roussel, 1996), Automated Mathematician (Lenat, 1976) and Circumscription (McCarthy, 1980) being especially notable contributions towards reasoning.

Built upon the developed systems and languages of inference were expert systems. Expert systems use the developed methods of representation and

reasoning to encode the knowledge of a field of endeavour that is held by a human expert. A set of encoded knowledge is known as a knowledge base. Arguably the most famous expert system is MYCIN (Shortliffe & Buchanan, 1975), which was able to identify bacterial infections and performed as well as or better than human experts (Yu *et al.*, 1979). Over the years, the knowledge contained within expert systems has become ever larger, leading to direct research into representation and inference systems that are capable of maintaining adequate performance. These expert systems that have vast number of rules are referred to as VLKBs – very large knowledge bases. These VLKBs look not just implement expert knowledge, but to all commonsense knowledge. By producing basic commonsense knowledge, this can augment any expert knowledge to provide an expert system able to respond to questioning in a much more robust manner (Lenat *et al.*, 1985). The most famous of the VLKBs is the CYC project (Lenat *et al.*, 1985; Lenat & Guha, 1990). CYC is an impressive multi-decade project that is attempting to encode all the commonsense knowledge held by a typical (western) person. To compare, CYC has millions of rules whereas MYCIN had in the region of 600 rules. Chapter one discussed a criticism this thesis holds on the approach taken by the CYC project, in relation to the claim that general intelligence-level behaviour can arise merely from encoding enough commonsense knowledge.

As the size of knowledge bases increased, arguably a third problem of commonsense reasoning has come to light. This problem is known as the knowledge acquisition bottleneck. The way that commonsense knowledge-bases have been traditionally built is through humans directly writing each individual rule in the format that the computer represents the knowledge in (or a format that is able to be directly and unambiguously transformed into the computer's format). This is a very laborious and costly approach, requiring highly skilled workers over an extended project to produce the numbers of rules needed. This problem has spawned research in finding ways to mitigate this problem. The work of this thesis can be argued to be a system that uses a method of reinforcement learning to allow for the acquisition of a limited sub-set of commonsense knowledge.

The remainder of this section will look firstly at research in knowledge representation that is pertinent to this thesis and then look at the various methods of knowledge acquisition that have been proposed. The problem of knowledge reasoning is out of the scope of this thesis because this thesis encodes any knowledge in the form of predicate logic – for which the methods of reasoning are well-known. Due to this, the reasoning problem is not discussed any further.

3.2.1 Knowledge Representation

The research into the representation problem has grown to be appreciably larger than the work into the reasoning problem. This is because it is sufficient to consider the reasoning problem in general terms but this is not as true

for the representation problem. This is due to the prevailing general method of representation – predicate logic – does not imply how specific types of knowledge can be encoded, such as temporal, spatial or social knowledge. Those first two examples, temporal knowledge and spatial knowledge are pertinent to this thesis and so the purpose of this section is to review the literature regarding those types of knowledge. Of the area of knowledge representation covered, Galton (2009) provides an especially articulate survey.

3.2.1.1 Temporal Knowledge Representation

Temporal knowledge is encoded in terms of the effects of actions or events on the current state of the world. Three predicate logic representations were considered for this project: the situation calculus (McCarthy, 1963), the event calculus (Kowalski & Sergot, 1986) and versatile event logic (Bennett & Galton, 2004). For reasons given later, a sub-set of the event calculus was used within this thesis.

The situation calculus, originally proposed by McCarthy (1963) has been extended in multiple ways by McCarthy, Reiter and others (McCarthy & Hayes, 1969; McCarthy, 1986; Reiter, 1991; Levesque *et al.*, 1998; Pirri & Reiter, 1999). The core of the representation is the $\texttt{result}\,(s, a)$ function, that gives the resulting world situation of the action (or event) a happening in the situation of the world s. Another main predicate is the $\texttt{trueIn}\,(s, P)$ predicate, that is true when P is true in situation s. This predicate gives a way to describe the current situation and place preconditions on the ability of the result function to change the current situation. If the preconditions are not met, no new situation occurs.

The event calculus was first described by Kowalski & Sergot (1986). This was later expanded on by Shanahan and others (Shanahan, 1999; Sadri & Kowalski, 1995; Miller & Shanahan, 1999). The Event Calculus has a couple of core predicates. The first of these is the $\texttt{holdsAt}\,(f, t)$ predicate which states that the fluent f is true at the time-step t. A fluent is a property that is allowed to change over time. Events are represented within the time step with three more predicates: $\texttt{happens}\,(a, t)$, stating that action (or event) a occurs at time-step t; $\texttt{initiates}\,(a, f, t)$ stating that fluent f is made true after action (event) a at time t; and $\texttt{terminates}\,(a, f, t)$ stating that fluent f is made false after action (event) a at time t. There is one particular extension to these core predicates that shall be used later. This is an alternate $\texttt{happens}\,(a, [t_1, t_2])$ predicate. This predicate is true when the action (event) a occurs over the duration of the interval starting at time t_1 and ending at time t_2. This predicate is used for actions and events that are not instantaneous.

Versatile event logic, the youngest of the three representations considered, was first proposed by Bennett & Galton (2004) with the aim of creating a single representation encompassing the advantages of the many representations of time and events. This is done through using a modal logic that forms a tree

of functions mapping time points to properties of the world that hold at that time point. This is then expanded upon with extra features such as event types.

Of these three representations, a sub-set of the event calculus was used in the development of this thesis. This is because while versatile event logic gives the most flexibility and power of expression, it has gained this flexibility by increasing the complexity with which facts are expressed. This would have increased the development time for an expressiveness that would have not been used. As the event calculus can be represented in versatile event logic, a more general variant of the proposed system could be implemented in the future within versatile event logic. The event calculus was also more suitable than the situation calculus for the proposed system due to the usage of video data. The situation calculus does not explicitly use discrete time-steps, however video data naturally has discrete time steps built-in in the form of frames.

A further highly influential piece of research on the representation of events concerns the relationship between events. When two events occur over an interval rather than happen at a single point in time, there can be 13 different distinctive relationships between the two event intervals. These relationships can be encoded as knowledge in relation predicates known as the Allen relations (Allen, 1983). Figure 3.1 shows these interval relations. While the relationship between two event intervals is a major part of this thesis, the Allen relations are not directly used. This is because this thesis is concerned with the existence of a temporal relationship rather than its precise nature. The existence of a relation is based on a threshold of the allowed time-gap between any two relations. If the gap is too large, then no relation exists. In terms of the Allen relations, this means that the `before` (A, B) relation and its inverse are effectively split into a `beforeNear` (A, B) relation and a `beforeFar` (A, B) relation, with their corresponding inverse relations. With these effective relations plus the other eleven relations, a relationship is considered to exist between two event intervals if the relationship is not the `beforeFar` (A, B) relation or its inverse.

3.2.1.2 Spatial Knowledge Representation

The naive approach to representing space within a commonsense knowledge system would be standard Cartesian geometry. However, this is not an ideal representation for any program with commonsense knowledge because it does not readily allow for any spatial knowledge of the system to be used for inference along-side non-spatial data. Because Cartesian geometry does not readily allow for inference along-side non-spatial knowledge, spatial knowledge is instead represented qualitatively. The basis for qualitative spatial reasoning is the relationship between spatial regions. There are two main schools of thought for how the relationships between regions should be qualitatively represented. The first school is based on the concept of intersection and the second school is based on the concept of connection.

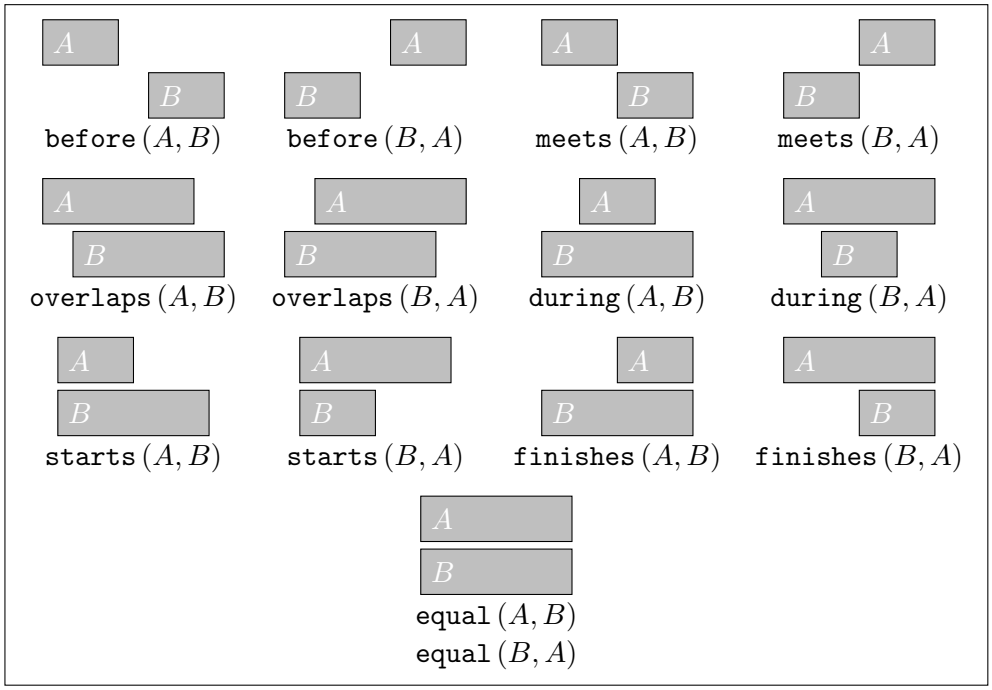

Figure 3.1: The thirteen Allan (1983) relations.

The first school of thought is based on the work of Egenhofer (1989; 1991). This school has three central models of spatial representation: the 4-intersecion model, the 9-intersection model and the dimensionally-extended 9-intersection model. The 4-intersection model (Egenhofer, 1989) considers a region to consist of its interior and its boundary. The relationship between two regions is then characterised by how the interiors and boundaries of each intersect – i.e. the boundary of region A with the boundary of region B, the boundary of region A with the interior of region B, the interior of region A with the boundary of region B and the interior of region A with the interior of region B.

This relationship of intersections is then expressed in a two-by-two matrix where each cell of the matrix contains a $\varnothing$ symbol or a $\neg\varnothing$ symbol. The $\varnothing$ symbol denotes there is no intersection for the given combination – i.e. the intersection of the set of points of the first region feature and the set of points of the second region feature is the empty set. The $\neg\varnothing$ symbol denotes there is an intersection for the given combination – i.e. the intersection of the set of points of the first region feature and the points of the second region feature is not the empty set.

To clarify this description, consider the example given in figure 3.2. Table 3.1 shows the 4-intesection matrices between region A and the other three regions. As can be seen, regions A and B only intersect at their boundary, regions A and C intersect at all four possible different intersections and region D intersects only with the interior of region A. Note that just because all the points of a region intersects with another does not mean that every region feature intersects.

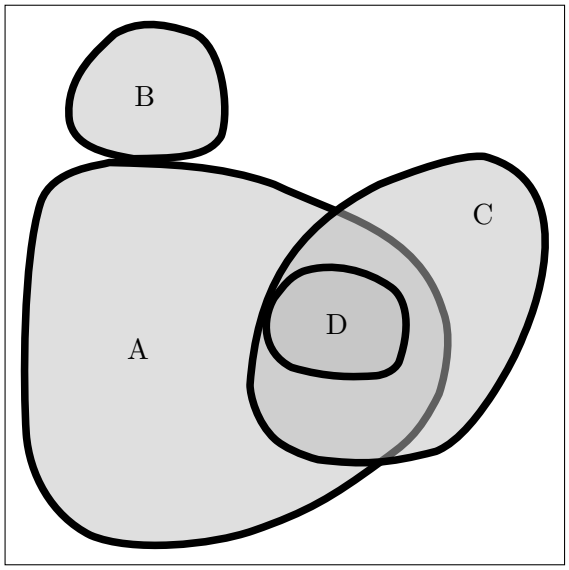

Figure 3.2: An example of regions with different relationships to one another.

$A - B$	boundary (B)	interior (B)
boundary (A)	$\neg\varnothing$	$\varnothing$
interior (A)	$\varnothing$	$\varnothing$

$A - C$	boundary (C)	interior (C)
boundary (A)	$\neg\varnothing$	$\neg\varnothing$
interior (A)	$\neg\varnothing$	$\neg\varnothing$

$A - D$	boundary (D)	interior (D)
boundary (A)	$\varnothing$	$\varnothing$
interior (A)	$\neg\varnothing$	$\neg\varnothing$

Table 3.1: Three examples of spatial knowledge represented as a 4-intersection matrix.

The 9-intersection model (Egenhofer, 1991) adds to the 4-intersection model by including a new feature of a region: the exterior of a region. The exterior of a region is all the points of the environment that are not a part of the interior or boundary of a region. The reason that there is a need for a 9-intersection model is to allow for objects in higher-dimensional space, or where the number of dimensions differs between two objects, such as capturing the relationship between a region and a line (Egenhofer *et al.*, 1993).

The dimensionally-extended 9-intersection model (Clementini *et al.*, 1993) adds to the 9-intersection model by replacing the $\varnothing$ and $\neg\varnothing$ symbols with the number of dimensions that each intersection has, using -1 to denote that there is no intersection, zero for a point-intersection, one for a line-intersection and two for a region-intersection and so on if needed for higher dimensions. This allows for yet further detail of how a region or other object relates to one another. For instance, consider two arbitrary lines on a plane. If they intersect on their interior (i.e. not at the end-points), in the 4-intersection or 9-intersection model this could mean either crossing at a single point or by sharing a line segment – there would be no way to disambiguate. In the dimensionally-extended 9-intersection model, the ambiguity would be removed, as a single point-crossing would be a zero-dimensional intersection and the shared line segment would be a one-dimensional intersection.

The other school of thought in qualitatively representing spatial knowledge is based on the concept of regions being connected. From this follows a set of predicates that form the Region Connection Calculus (RCC). The founding and canonical version of this calculus has eight relations and is therefore referred to as RCC-8 (Randell *et al.*, 1992). The relation of two regions being connected is taken as primitive. From this, a part relation can be defined by stating that every region connected to a part of a region is connected to the whole of that region. This in turn allows for an overlap relation to be defined by stating that two regions that overlap each have a part that is a part of the other region. These three relations, the connected relation, the part relation and the overlap relation can then be used to derive the final eight relations that are jointly exhaustive and pairwise disjoint (JEPD – i.e. there is no configuration of two regions that isn't described by a relation and there is no configuration of two regions that is described by more than a single relation). Table 3.2 lists the predicates of the relations and their description. Figure 3.3 depicts each of the relations and shows the transitions between the relations that are possible without needing to transition to another relation first (known as the conceptual neighbourhood).

From RCC-8, there are a number of variations that can be created. The first of these variants is known as RCC-5 and was first described by Bennett (1994). The change from RCC-8 is that there is no discrimination made to whether the boundaries of the two regions are in contact. The consequence of this is that there is not discrimination between tangential proper parts and non-tangential proper parts and also no discrimination is made between

Predicate	Description
DC (A, B)	A is disconnected from B.
EC (A, B)	A is externally connected to B.
PO (A, B)	A partially overlaps B.
EQ (A, B)	A is equal to B.
TPP (A, B)	A is a tangential proper part of B.
TPPi (A, B)	B is a tangential proper part of A.
NTPP (A, B)	A is a non-tangential proper part of B.
NTPPi (A, B)	B is a non-tangential proper part of A.

Table 3.2: The RCC-8 relation predicates.

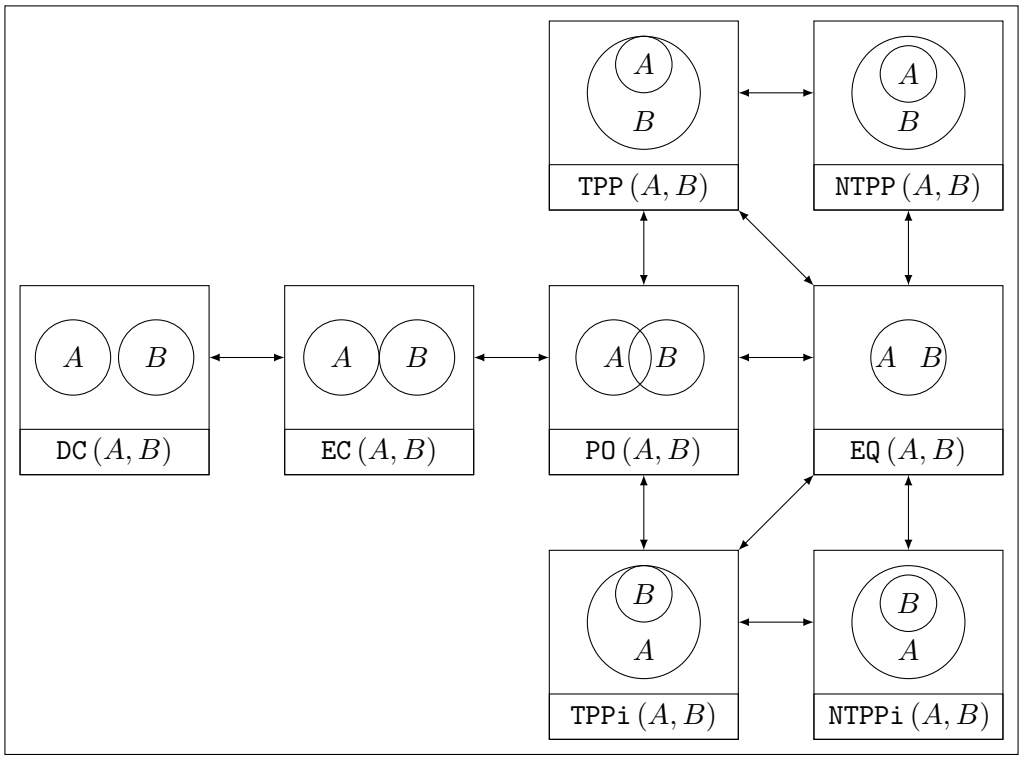

Figure 3.3: The RCC-8 relations in their conceptual neighbourhood.

regions that are externally connected and regions that are disconnected. This change makes it possible for any automated reasoning done using the representation computationally easier than using RCC-8, i.e. the representation is able to be used in a propositional logic rather than first-order logic. Table 3.3 lists the predicates of each relation of RCC-5 along with a description of the relation.

Predicate	**Description**
`DR` (A, B)	A is discrete from B
`PO` (A, B)	A partially overlaps B
`PP` (A, B)	A is a proper part of B
`PPi` (A, B)	B is a proper part of A
`EQ` (A, B)	A is equal to B

Table 3.3: The RCC-5 relation predicates.

The next variant of the region connection calculus is known as RCC-3 and was originally proposed by Santos & Shanahan (2002). This variant is used where the uncertainty of the knowledge means that when regions overlap, the two regions cannot be reliably distinguished from each other. This means that there is no discrimination made between the part relations, the equality relation and the partial overlap relation. Table 3.4 lists the predicates of each relation of RCC-3 along with a description of each relation.

Predicate	**Description**
`DC` (A, B)	A is disconnected from B
`EC` (A, B)	A is externally connected to B
`CO` (A, B)	A is coalescent with B

Table 3.4: The RCC-3 relation predicates.

Variants of the region connection calculus do not just reduce the number of relations available. There is a variant known as RCC-23 by Cohn *et al.* (1997). The RCC-23 variant introduces the concept of the convex hull of a region. This allows for the definition of relations that describe the relationship between concave regions. Due to the complexity of the theory and that it is out of the scope of this thesis, further details of this variant shall not be discussed.

Of the representations presented, this thesis makes use of the RCC-3 representation. This is primarily due to practical constraints that have been caused through other choices made in the development of the thesis. There are two constrains that led to this choice. Firstly, and most importantly, the input of the system described in chapter four is track-box data where it is not possible to distinguish between regions that overlap by a significant degree. Secondly, one of the modules of the system is based upon earlier work by dos Santos *et al.* (2009) that uses RCC-3 for similar reasons.

For a more principled choice, the variants of the region connection calculus are deemed by this thesis to be preferable to the variants of the intersection

model. This is due to the criticism that many of the configurations of the intersection model are impossible to create – for instance the interiors of two objects cannot intersect without at least one of the boundaries of the two objects intersecting with the other object in some way. In addition, some of the intersection configurations are equivalent (Zlatanova *et al.*, 2004, p. 425). There are criticisms of the region connection calculus, such as it requiring that both regions are of the same number of dimensions (Galton, 2009) and that in a discrete space, the calculus has the problem that the atomic regions are parts of their complements, though this issue is only a boundary case and there exists work that tackles this issue, such as that by Roy & Stell (2002).

Both the intersection and connection representations only concern the topological relationship between regions. There is also a number of ways in which the directional relationship between regions can be represented, three of which will be discussed in this section. All representations of direction are relative to the position of a reference object, point or direction. The first two representations were proposed by Frank (1992). The first representation, shown in figure 3.4a, is known as the cone representation. The cone representation divides the space around the reference point diagonally into quarters, creating four qualitative directions corresponding to the four primary compass directions. The second representation, shown in figure 3.4b, is known as the projection representation. The projection representation divides the space around the reference point into nine squares, one for each of the eight main compass point and a central region which Frank (1992) called the neutral zone.

The final representation covered, shown in figure 3.4c, was introduced by Freksa (1992) and extended by Scivos & Nebel (2001). The representation, known as the double-cross calculus, is based upon an observer's position and observation direction, with the observer being situated at the base of the arrow in figure 3.4c and focusing towards the head of the arrow. This divides the space around the observer into three horizontal parts and five depth parts. The three horizontal parts are: left of the observer (L), straight-on with the observer (S) and right of the observer (R). The five depth parts are: "in-front" of the focus point (F), perpendicular to the focus point (P), in-between the observer and the focus point (centre - C), in-line with the observer (L) and behind the observer (B).

Of the direction representations presented, the closest this thesis uses is that of the projection representation. This is again primarily for practical reasons related to the use of a kernel (box) tracker representation for the input to the system presented in chapter four and the fact that one of the modules is based on the work of dos Santos *et al.* (2009). From a more principled viewpoint, it appears that the strengths of the three representations very much depend on the task at hand. If the observer and the viewpoint are concepts that are a part of the task at hand then the double-cross calculus is by far the best choice due to its expressiveness. On the other hand, if in the task at hand there was no usable reference vector, then the representation is not usable. Of

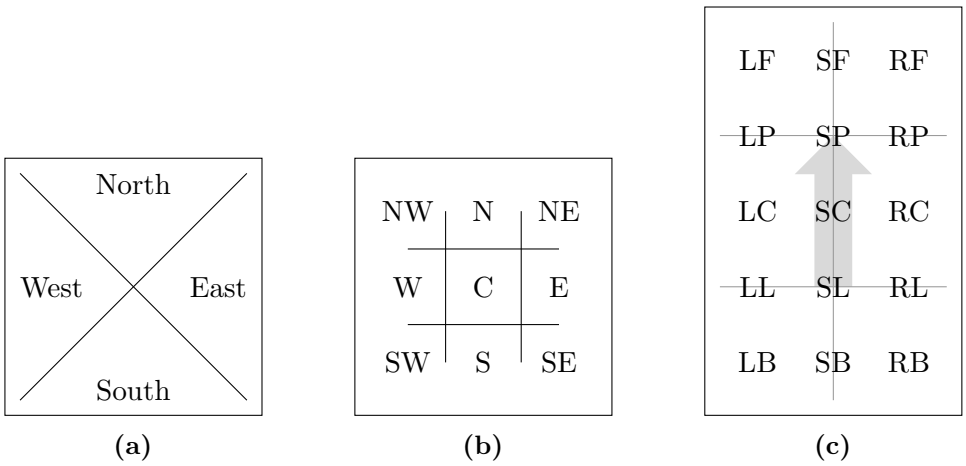

Figure 3.4: Three possible representations of direction.

the two frank representations, the projection representation looks to be the more general of the two, as it takes into account the size of the reference object, rather than assuming that it is suitable for the reference location to be represented as a point.

3.2.1.3 Spatiotemporal Knowledge Representation

There has been a wide range of work done looking at creating a qualitative representation of movement and spatial change over time, much more than could be covered in this subsection. Galton (2000) has produced an authoritative account of the topic. There appears to be two approaches to the problem of combining space and time. The first approach is to use an existing theory of space or time and either extend it into the other or combine it with another existing theory. The second approach is to develop a new calculus specifically for the goal of representing spatiotemporal knowledge.

The first approach of combining existing theories of time and space has produced work such as in combining Allen's interval calculus with region connection calculus (Gerevini & Nebel, 2002; Bennett *et al.*, 2002). This is a natural approach to the problem, given that the region connection calculus can be seen to be a two-dimensional extension of Allen's interval calculus. Another piece of work using this approach is the qualitative trajectory calculus (Van de Weghe *et al.*, 2006) which uses the region connection calculus but extends this using the concept of the absolute and relative velocities between objects.

In the second approach, that of developing a new calculus, there has been work such as that by Stell & del Mondo (del Mondo *et al.*, 2010; Stell *et al.*, 2011) where change over time is represented as a graph relating entities in one time to the entities they came from and to the entities they became. Another work in this category is to define a four-dimensional spatial calculus (for instance, based upon the notion of connection) and then allow for one of the dimensions to be interpreted as time (Muller, 2002; Stell & West, 2004).

3.2.2 Knowledge Acquisition

As previously described in this section, the commonsense knowledge present in a knowledge base is traditionally acquired by computer systems through a process of knowledge engineering. This process is tedious and costly. There have been numerous approaches to attempt to deal with this problem. These approaches fall into two general categories. The first of these categories is the use of user interfaces that allow the general public to add to the knowledge of the system. The second of these approaches is to use natural language processing systems to learn facts from web pages. Both of these categories are reviewed in turn in this subsection, and then the relationship of this thesis to knowledge acquisition shall be discussed.

3.2.2.1 Knowledge Acquisition Interfaces

The research into knowledge acquisition interfaces focuses on attempting to allow untrained or minimally trained users to encode and express commonsense knowledge. By reducing the skill-level needed to encode knowledge, it allows for the acquisition task to be distributed among a larger workforce, such as volunteers using a collaborative system over the world-wide web. This reduces the problem of the knowledge acquisition bottleneck in three ways: firstly by reducing the cost of the workforce required, secondly by increasing the speed at which a piece of knowledge can be encoded and thirdly by allowing the task to be distributed among a larger number of people.

One of the first and by-far the largest contribution to layperson knowledge acquisition is that of the Open Mind Common Sense project, which directly aimed to create a knowledge base with a layperson knowledge acquisition method. The earliest and primary foundation work to do with this project is that by Singh *et al.* (Singh, 2002; Singh *et al.*, 2002; Singh & Barry, 2003). The key to this work is that it uses the English language itself as the internal representation for the knowledge of the system. Instead of using a machine-readable representation, the project has created rules of inference that work on the English language. Therefore, in this project, people are asked to enter simple facts in English such as "A cat is a mammal". When inference on the knowledge is conducted, the natural language parsing techniques match various templates such as "X is a Y" and when a template is matched, it allows for various rules of inference to be used, such as disambiguating a rule, paraphrasing the rule, splitting and merging of facts and other heuristics.

The Open Mind Common Sense project was then expanded by Liu & Singh (2004), introducing the ConceptNet. ConceptNet itself was later expanded upon by Havasi, Speer and others (Havasi *et al.*, 2007; Speer, 2007; Speer *et al.*, 2009) This work applies the various relationships in the Open Mind Common Sense knowledge base into a network of concepts. In the network, the nodes represent compound concepts such as "full stomach" and "eat breakfast". The connections in the network represent the English lan-

guage relationship. The relations are the parsing templates, so the nodes "full stomach" and "eat breakfast" have the connection labelled "effect of". By representing the relationships as a network of concepts, it allows for easier computation of reasoning over multiple connections, such as matching patterns of relationships for the creation of analogies and quick deduction based upon the transitive nature of some relationships such as "is a". One extension to this by Speer is to ask the user yes or no questions based on distant relations found within the network such as "Would you find shampoo in the living room?" based on both being found within a house. This allows for new relationships to be found that the user may not think to write down.

A further contribution from the Open Mind Common Sense project is the work by Gupta & Kochenderfer (2004). This work extended the base of the project in two main ways. Firstly, the work focused on indoor commonsense objects, by limiting the scope of the knowledge it was hoped that it would allow for a denser level of knowledge. Secondly, this work looked at the inference of actions using the English knowledge representation.

There has been some work that is not directly a part of the Open Mind Common Sense project but based upon it. The first of these is the system known as Learner (Chklovski, 2003; Chklovski & Gil, 2005). The system uses a small set of initial seed statements and the user selects an object to discuss. The Learner system then uses the properties of similar concepts as questions to whether they apply to the selected concept. In this way, new commonsense knowledge is prompted and gained. The work on Learner can be seen to be a prototypical version of ConceptNet.

The second piece of work in the area of using an interface for layperson knowledge acquisition is the Verbosity game by von Ahn *et al.* (2006). This presented the task of creating new knowledge in the form of a computer game. In the game, two people play cooperatively with each other. In turn, one person selects a word to be guessed and the other has a predetermined set of questions to ask about that word in a "fill in the blank" style. The person who initially selected the word then fills the blank in but must not use the selected word. Points are scored if the guesser guesses the word. Naturally, the questions that are asked are in the knowledge format used by the Open Mind Common Sense knowledge base and so each statement created by the game could be added to the knowledge base, if not already present.

The third piece of work related to the Open Mind Common Sense project is the work of Kuo & Hsu (2011). This work looked at creating a Chinese language version of the Open Mind Common Sense project, leveraging the English language version to aid the knowledge acquisition process. Kuo, Hsu & Shih (2012) more recently did a piece of work in the area looking at using content from social networks to increase the size of the knowledge base.

While the work surrounding the Open Mind Common Sense project represents the bulk of the work done in the creation of interfaces for layperson commonsense knowledge collection, there are other pieces of work in this area.

The earliest of which is by Witbrock *et al.* (2005). In their work, they review some methods for layperson knowledge acquisition that generate knowledge for the CYC knowledge base. The work demonstrates four graphical user interface systems that allow the user to input different kinds of commonsense knowledge. The first is an interface that takes simple natural language sentences and splits the individual facts into key-value relationships based on a chosen concept, further key-value relationships are then prompted for, based on concepts that have a similar set of relationships. The second interface allows for the user to match subjects to objects of a given type. The user is given a set of subjects and objects that have been observed to appear together on a web page and the found subjects and objects satisfy various filtering criteria. The third interface presents hypothesised statements based on abducing[2] from the existing knowledge within the knowledge base and then presents the statement to a user. The user then rates the statement for comprehensibility, appropriateness, truth, interest value and plausibility. The final interface uses inductive logic programming to create generalised rules that are then presented to the user with examples and the user is expected to state whether the rule is correct.

3.2.2.2 Knowledge Acquisition through Natural Language Processing

While at least some of the methods of layperson knowledge acquisition interfaces use natural language processing to process the input of the system, this was only for simple single-line facts. This section looks at the approach of using large natural language corpora from sources such as web pages to obtain commonsense knowledge.

The first work of this sort is by Gao & Sterling (1997) where a limited handcrafted knowledge base was used along with a natural language processing system. The knowledge base guided the processor to allow for further knowledge to be acquired. The study was very limited to the narrow use-case of understanding estate agent advertisements. The next work found that looks at mass corpora knowledge acquisition, by Wyatt *et al.* (2005) uses descriptions of activities from the web to recognise the activity from RFID sensor data.

Matuszek *et al.* (2005) produced an important study into using web corpora for commonsense knowledge acquisition. That study looked at the methods that are used to populate CYC with commonsense knowledge extracted from web pages. As with the earlier work by Gao and Sterling, the CYC knowledge base itself was used to aid the parsing process, to both generate query and check the consistency of the results. The search was performed through the Google search engine and the final knowledge rules from the results of the

[2]Abduction is a mode of inference where the agent attempts to infer the most likely explanation for an observed consequence.

parser were also verified by searching for an English language version of the rule.

Another study in using web pages to acquire commonsense knowledge is by Hadidi *et al.* (2010). In this study, the Simple English Wikipedia was parsed into sentences. Of those sentences, those that followed the pattern "noun phrase – verb phrase – noun phrase" were extracted. Those sentences were then used to form a relational network with each verb as the relation type.

The final study looked at in this section, by Mancilla-Caceres & Amir (2010; 2011) combines both approaches, a user interface for laypeople and the use of a large natural language corpus. The approach used was to provide a computer game that requires the user to classify whether a commonsense knowledge rule makes sense and is true. The candidate rules were taken from Wikipedia in the same manner as the study by Hadidi *et al.* (2010). In their system, a candidate rule was randomly selected either from the existing known rules or from a list of previously used rules. Two human game players were each asked whether the selected rule is true, false, nonsense, or not known. A third computer player made the same choice, based on the previous answer if it is a previously used question, or selects "not known" if the answer is a known answer. After all players had selected an answer, the human players were asked to decide which of the other two players was a human based on what answer the other two players gave, as a form of pseudo-Turing test. Points were awarded if the guess was correct and lost if the guess was incorrect. The authors argue that the inclusion of a computer player stops human players from being able to cheat by agreeing a fixed strategy and so not necessarily provide correct answers, as only when the players adopt the strategy of giving the correct answer can the players have a chance of knowing which player is the computer.

3.2.2.3 Knowledge Acquisition in Relation to this Thesis

The system proposed by this thesis does not fall into either of the existing categories. Instead it learns by observing the environment to accumulate patterns of events. These event patterns are encoded in a predicate logic in such a manner that they are usable for inference. For instance, if the system learns that a ball moving upwards will lead in short order to that same ball moving downwards, then if the existing knowledge base contained facts such as "people sometimes catch moving balls that move within their reach", then the system could possibly qualitatively infer the prediction that a ball moving in an arc towards a person may be caught, even though the ball is currently moving away from the person in the vertical axis. Admittedly, this is a rather tortured example due to the current limitations of what can be learned by the system presented. However, this does not discount the fact that the knowledge being learned is able to be used within a reasoning context and so it is argued that this system can be seen to be a prototypical knowledge acquisition system.

In the literature, only a single discussion was found of a system that looks at learning commonsense knowledge from purely observing the environment. This discussion was of the BabyExp project by Poesio *et al.* (2010; 2011). This was a voluntary research project that last reported its progress in 2011 with an unknown status after this point. The project reports stated that the project aimed to produce work that looked to analyse an automated transcription of an audio and video recording of the first three years of a baby's life. From this the group wished to develop algorithms that acquired commonsense knowledge from the transcription by attempting to exploit the same "training data" that a child uses to learn its linguistic knowledge. The status of the project in the latest report had carried out an initial look into the automated transcription of the video data. At that point, the project had not produced any methods to learn from the transcribed data.

3.3 Reinforcement Learning

The form of machine learning known as reinforcement learning attempts to learn a method of selecting actions such that the reward received is maximised. The field has links across artificial intelligence and computer science in general, from dynamic programming to neural networks to planning.

Reinforcement learning has in some form been a part of the field of artificial intelligence from early on in its history. Minsky (1952) discussed a form of reinforcement learning, just two years after Turing (1950) asked "Can Machines Think?" Minsky later discussed secondary reinforcement in his PhD thesis (Minsky, 1954). A notable early piece of work is Samuel's study of checkers-playing programs (Samuel, 1959). The discussion of these programs make reference to a "reward-and-punishment routine" and looked at storing the score produced by a given move for each board position encountered – which Samuel called "rote learning" and can be seen as a very basic temporal difference method – a form of reinforcement learning.

Later on, as discussed earlier in this chapter, Sutton & Barto produced two models of classical conditioning based upon the concept of credit assignment over time (Sutton & Barto, 1981, 1987), inspired by Klopf's (1972) work on conditioning and his general cybernetic theory of heterostasis. One of these two models, the temporal-difference (T.D.) model, was later abstracted away from its basis in classical conditioning to form the reinforcement learning method known as T.D. learning (Sutton, 1988). This was then later expanded upon by Watkins to produce a system known as Q learning (Watkins, 1989; Watkins & Dayan, 1992). These two methods, T.D. learning and Q learning form the basis of modern reinforcement learning research. Both of the two methods will be discussed in more detail later in this section.

For an overview of the field, the survey by Kaelbing *et al.* (1996) and the highly influential book by Sutton & Barto (1998) both provide a good grounding. Due to the growth of the field, more recent surveys focus on

different aspects of reinforcement learning, though Gosavi (2009) has presented a tutorial survey that reviews an updated core of reinforcement learning from a dynamic programming perspective. The more specific surveys cover topics such as multi-agent reinforcement learning (Buşoniu *et al.*, 2008), learning from demonstration / apprenticeship learning (Argall *et al.*, 2009) and transfer learning (Taylor & Stone, 2009). Not being directly relevant to this thesis, these different extensions are not covered, though two extensions shall be briefly covered later in this section.

3.3.1 The Reinforcement Learning Problem

A concise yet highly instructive definition of the reinforcement learning problem is provided by Sutton & Barto:

> *"Reinforcement learning is about learning from interaction how to behave in order to achieve a goal. The reinforcement learning agent and its environment interact over a sequence of discrete time steps. The specification of their interface defines this particular task: the actions are the choices made by the agent; the states are the basis for making the choices; and the rewards are the basis for evaluating the choices. Everything inside the agent is completely known and controllable by the agent; everything outside is incompletely controllable but may or may not be known. A policy is a stochastic rule by which the agent selects actions as a function of states. The agent's objective is to maximise the amount of reward it receives over time."*
>
> *(Sutton & Barto, 1998, p. 81)*

This definition covers all almost all of the basic terminology of reinforcement learning. As described in the definition, the task at hand is for an agent to learn about how its actions affect the environment that the agent is in. When the program starts, the agent is told the environmental state that it is in, but may know nothing else about the environment. At each state, the agent has a set of actions it can perform, from which it must select a single action to perform on the environment. Once an action is performed on the environment, the agent is given two pieces of feedback information: a reward and a new description of the state of the environment. The reward can be positive (a "reward"), negative (a "punishment") or zero valued (a neutral state). The goal of the agent is to maximise the cumulative reward. This is achieved by a policy, which decides which action to take given a particular state of the environment. At its most basic, a policy is a list of instructions stating "if you are in state A then do action B". In a more general fashion, a policy can be probabilistic, where for each action that can be taken in a given state, there is a probability that that action will be taken. The function $\pi(s, a)$ denotes the probability that action a will be taken in state s for the policy π. The task

of reinforcement learning is then one of finding the best policy, i.e. the set of probabilities that maximises the cumulative reward.

Initially, since the agent knows very little about the environment, the actions have to be selected randomly. However, as different actions are selected and their reward is revealed, the policy can be progressively changed so that the actions that deliver the best reward are selected. If the world has been completely explored, then the policy would be able to select the sequence of actions from any starting point that gives the best possible reward. The policy that gives this best possible reward is referred to as the optimal policy. The optimal policy is denoted as $\pi^\star$ and the function $\pi^\star(s, a)$ denotes the probability that action a will be taken in state s for the policy $\pi^\star$.

Consider the case, that while exploring, the agent happened by accident to come across the best sequence of actions. In this case there would be little need to explore further as any further exploration would not improve the policy, so therefore it is not necessary to exhaustively search the entire environment to learn the best policy. However, this leads to a dilemma. The agent does not know when it has arrived at the best policy, so unless there has been an exhaustive search of the environment, the agent cannot be completely confident that there is not a better policy, hidden in the unexplored parts of the environment. The dilemma is that the agent can choose to exploit the current best policy or choose to continue to explore in hope of finding a better policy. If it already knows the best policy, then any time spent exploring is wasted and should have been spent exploiting the current policy. This is known as the exploration-exploitation dilemma and is central to the reinforcement learning problem.

3.3.2 Value Functions

The way to store and compute the best policy is through a value function, of which there are two possible types, a state value function and a state-action value function. The state value function is usually denoted as $V(s)$ where s is the state and is used when the rewards of the environment need to be predicted, but the learning system is not in control of action selection. The state-action value function is usually denoted as $Q(s, a)$ where a is the action and is used when the action needs to be selected by learning system. A state value function assigns a value to measure how beneficial it is to be in a particular state. A state-action value function assigns a value to measure how beneficial it is to select a particular action in a particular state.

The measure of the benefit of a state or state-action pair is based on the need for the agent to maximise the cumulative reward over the life of the agent. This means that the value assigned needs to include not just the immediate reward, but the cumulative reward that would be achieved if starting with that state or state-action pair and following the current policy. As the value functions have to be defined in relation to a particular policy, the function can instead be written as $V^\pi(s)$ and $Q^\pi(s, a)$ respectively, where π refers to the

policy that the value function relates to. Where the policy refers to the optimal policy, the functions are usually written $V^{\star}(s)$ and $Q^{\star}(s,a)$ respectively.

The total reward for an agent operating under a particular policy can only be known after the life of the agent. This can be problematic if the agent is expected to improve during its lifetime. Policy improvements that occur within the lifetime of an agent are dealt with by discounting the sum of rewards over time. The time-discounted future cumulative reward is known as the expected return. The reinforcement learning methods can therefore be described in terms of attempting to estimate this expected return for the value function, as estimating the expected return would allow for selection of the best policy. Now that the notion of discounting the reward has been discussed, equation 3.14 and equation 3.15 define the expected return for the state and state-action value functions respectively, where r_t denotes the reward at the t^{th} time-step, s_t denotes the state t^{th} time-step, a_t denotes the action taken at the t^{th} time-step, γ is the time-discount value and $E_{\pi}\{\}$ denotes the expected value if the agent follows the policy π.

$$V^{\pi}(s) = E_{\pi}\left\{\sum_{k=0}^{\infty}\gamma^{k}r_{t+k+1}\middle|s_t = s\right\} \tag{3.14}$$

$$Q^{\pi}(s,a) = E_{\pi}\left\{\sum_{k=0}^{\infty}\gamma^{k}r_{t+k+1}\middle|s_t = s, a_t = a\right\} \tag{3.15}$$

3.3.3 Controlling Exploration

Now that the value functions have been described, the reason why policies have to be probabilistic can be discussed. The reason the action selection of a policy is defined to be probabilistic is because of the exploration-exploitation dilemma. By having a probability that the best-known action will not be taken, it allows for a reinforcement learning system, through the current policy, to allow for exploration of the environment. When the reinforcement system changes the current policy, it can change the probabilities and so can control exploration over time. There are a wide range of strategies to control exploration, some of which were reviewed by Thrun (1992). Two widely-used strategies for controlling exploration are known as the ϵ-greedy method and the Softmax method. Note these methods both depend on the value function and are not a substitute for it, nor would any method for balancing exploration and exploitation make sense without it, otherwise the method would not know what the exploitation choice was.

In the ϵ-greedy method, the best known action is selected most of the time but occasionally, with probability ϵ, a different action is selected from the remaining actions (with uniform probability between the remaining actions). Equation 3.16 describes the probability that action a would be chosen under the ϵ-greedy method, where n is the number of actions available to the agent at state s and ϵ is the probability that the best known action will not be taken. Sutton & Barto (1998, p. 48) attribute the method to Watkins (1989).

$$\pi(a,s) = \begin{cases} 1-\epsilon & \text{if } a = Q^{\pi}(s,a') \\ \frac{\epsilon}{n-1} & \text{otherwise} \end{cases} \tag{3.16}$$

In the Softmax method, the actions are selected with a probability that is dependent on their current expected reward, with the largest probability being assigned to the best known action and the lowest probability to the worst action. The probability distribution for the Softmax method is the Boltzmann distribution. Equation 3.17 describes the probability that action a is selected, where n is the number of actions available to the agent at state s and τ is the temperature. Temperature values tending towards positive infinity tend towards all the action probabilities being equal, temperature values tending towards zero tend towards always choosing the action with the largest expected return. The Softmax rule was first proposed by Luce (1959).

$$\pi(a,s) = \frac{e^{Q^{\pi}(s,a)/\tau}}{\sum_{a'=1}^{n} e^{Q^{\pi}(s,a')/\tau}} \tag{3.17}$$

3.3.4 Stochastic Results and Markov Decision Processes

There is a final complicating factor that is part of the reinforcement learning problem but was not discussed before this point to avoid making the discussion harder to follow. This is the fact that when an action is selected, either by the learning system or otherwise, the following state is not always the same. Instead, the state following from the selected action is randomly decided from a probability distribution over a set of states. The set of states is usually a small sub-set of the total states so the randomness is still constrained. The stochastic nature of the resultant states consequently affects the reward for a particular action, as the reward is given upon entry into a state from the chosen action. Because the reward is based both upon the action and the resultant state, it is possible for two separate actions to lead to the same state but have different rewards. An example where this may apply is if the agent is rewarded by trying to go up a hill, but sometimes slips so fails to go up but is still rewarded for trying, whereas choosing to go down could lead to the same state but wouldn't be rewarded.

By making the system stochastic, it changes the requirements for the approach that must be taken for by a reinforcement system, but does not change the nature of the value functions, and equation 3.14 and equation 3.15 still apply. The nature of the exploration-exploitation dilemma and the methods mentioned also still apply. The reason that the value functions are not affected by the inclusion of this stochastic behaviour is because the stochastic behaviour can be fully contained within the expected value (denoted by $E_{\pi}\{\}$ in equation 3.14 and equation 3.15). The *actual* sum for the time-discounted awards is not changed, as there is still only one reward and one successor state given after an action selection.

This system where the environment's successor state and reward response to an agent's actions is stochastic can be represented by a mathematical framework known as a Markov decision process (MDP) (Bellman, 1957).

A Markov decision process can be represented as a 4-tuple:

$$\left(S, A, P\left(s, a, s'\right), R\left(s, a, s'\right)\right)$$

Where:

- The symbol S denotes the set of states.
- The symbol A denotes set of actions.
- The function $P : S \times A \times S \rightarrow [0, 1]$ denotes the probability $p \in [0, 1]$ that taking action $a \in A$ from state $s \in S$ will lead to state $s' \in S$.
- The function $R : S \times A \times S \rightarrow R$ denotes the immediate reward $r \in R$ for taking action $a \in A$ from state $s \in S$ that led to state $s' \in S$.

A MDP can be represented diagrammatically as a directed graph with two kinds of node: a state node and an action node. Edges from state nodes lead to action nodes and edges from action nodes lead to state nodes. The edge from action nodes to state nodes is labelled with the probability of that transition occurring along with the reward given when that transition is used. Figure 3.5 shows an example graph for a MDP with two states and two actions. In figure 3.5 the optimal policy is to always select action X. Note that even when the transitions are defined stochastically, a MDP can still contain deterministic elements, such as the (A, Y) state-action pair of figure 3.5.

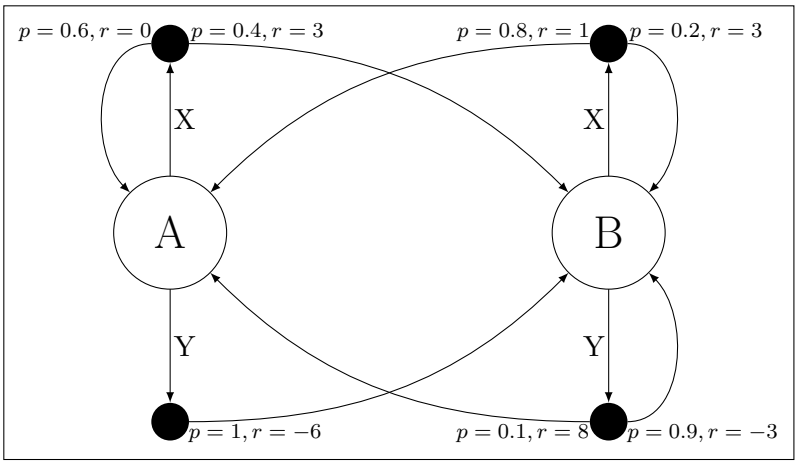

Figure 3.5: A generic example of a Markov Decision Process. The empty nodes (A and B) are state nodes. The filled-in nodes are action nodes. X and Y are the possible actions. The value p is the transition probability. The value r is the reward given for that transition.

By expressing the reinforcement learning problem in terms of finding an optimal policy over an MDP, the value functions can be expressed even more precisely. This is done through defining the value function in terms of the

policy action selection function, the transition function of the MDP and the reward function of the MDP. The equations that do this are known as the Bellman equations (Bellman, 1957). The Bellman equations are out of scope for this thesis as they are needed to understand neither the remaining discussion of reinforcement learning, nor the relationship between this thesis and the reinforcement learning literature, nor the system presented by this thesis.

3.3.5 Approaches to the Reinforcement Learning Problem

There are three main approaches to the reinforcement learning problem: The dynamic programming approach, the Monte Carlo approach and the temporal difference approach. One of the central differences is that in the dynamic programming approach, the MDP of the environment is assumed to be known, and so examples from actual experience are not needed. The other two approaches do not assume that the MDP is known, and instead rather try to learn the optimal value functions from actual experienced cases.

The dynamic programming approach treats the reinforcement learning problem as an optimisation problem. As the approach already has complete knowledge of the environment, there is no need for the policy to be stochastic. Instead the policy is deterministic, i.e. each environmental state has exactly one action choice. The goal in this optimisation problem is to produce an optimal policy. This is done by evaluating each policy and then evaluating variants of that policy by selecting a different action at a given point and but keeping the remainder of the policy the same. If the end value is increased by adopting the new action then the policy is changed to include this action. This feedback process of evaluating the policy and then selecting a local action that improves the evaluation is known as policy improvement. The key then to the dynamic programming approach is the evaluation of a policy. One approach is to treat the MDP as a system of simultaneous equations. The other method is to change the relevant Bellman equation to be an iterative update rule and iteratively change the value function until it converges (Bellman, 1957).

The Monte Carlo approach is that an agent is given a stochastic policy which it must follow for its entire life. At the end of the life of each agent, the cumulative reward is added to each and every state or state-action pair that the agent encountered during its lifetime. The value for each state or state-action pair is then the mean of the cumulative reward over the lifetime of many agents. By the law of large numbers, the value for each state or state-action pair will converge to its true value (Michie & Chambers, 1968).

The temporal difference approach (Sutton, 1988; Sutton & Barto, 1998) is similar to the Monte Carlo approach in that it uses actual experience to estimate the value function. However, unlike the Monte Carlo approach, the temporal difference approach updates the value function within the lifetime of each agent. One approach to this is done by only calculating the value of the a given state or state-action pair that was visited based on the value of the next state or state-action pair that was visited, plus the immediate reward that was

received for transitioning between the two cases. Initially, the value function will only take into account immediate rewards, but as the agent re-visits states or state-action pairs, and over the lifetimes of many agents, the value function will converge to be globally optimal rather than locally optimal.

Being born out of conditioning, the temporal difference approach is the closest of the three approaches to this thesis. Because of this, the remainder of this section on reinforcement learning will focus on this approach.

3.3.6 Temporal-Difference (TD) Learning

The TD learning method was developed out of the TD model of classical conditioning, as discussed earlier in this chapter. The method abstracts away from the concept of stimuli and instead looks purely at predicting future rewards given the current state. The temporal difference learning method presented here is the same as that presented by Sutton & Barto (1998), which can be considered to be the canonical version of the method.

In the terms of the notation introduced in this section on reinforcement learning, equation 3.18 gives the amount a state value will be updated by. In equation 3.18, α is a learning rate constant, $e(s_t)$ is the eligibility trace (explained below) and the remaining symbols are as were previously defined.

For comparison, a slightly-rearranged version of the main equation of the TD model of classical conditioning (equation 3.13) is presented in equation 3.19. In that equation, ΔV_i is the change in association strength for the i^{th} conditioned stimulus; $\overline{X_i}$ denotes the eligibility trace for the i^{th} conditioned stimulus; λ_{t+1} is the magnitude of the conditioning strength at time $t+1$; α_i denotes the conditioned stimulus-specific learning rate; β is the unconditioned stimulus-specific learning rate; γ denotes the imminence weighting and $\overline{V_t}$ is the prediction made of the magnitude of the unconditioned stimulus at time t.

$$\Delta V(s_t) = \alpha e(s_t)\left(R(s_t, a, s_{t+1}) + \gamma V(s_{t+1}) - V(s_t)\right) \tag{3.18}$$

$$\Delta V_i = \alpha_i \overline{X_i} \beta \left(\lambda_{t+1} + \gamma \overline{V_{t+1}} - \overline{V_t}\right) \tag{3.19}$$

In looking at the similarities and differences between the two equations, it is noticeable that there are many analogies between the two equations. The prediction of the value remains the same, with a discount factor; the immediate magnitude of the unconditioned stimulus is now the immediate reward; there is a value for the eligibility of the association and there is a fixed learning rate constant. The one place where there is not an analogy between the two equations is that the reinforcement learning method does not have a specific learning rate for each state – this is probably because such a learning rate would make little sense in the context of the reinforcement learning problem.

Algorithm 3.1 lists the full `TD`(λ) algorithm, applying equation 3.18. The listed algorithm is the same as that described by Sutton & Barto (1998,

p. 174)[3]. The way the algorithm works is by back-propagating rewards through the chain of states that were experienced within the lifetime of the agent as the rewards are experienced, discounting by γ at each step. The eligibility trace for each state ($e(s)$) controls this discounting by being multiplied by the discount factor each time the state is updated. This reflects that the state is one step further in the past each time the agent goes to a new state. Note that in line nine, the eligibility trace is incremented by one, this is so that if a state is visited more than once in the lifetime of the agent, that state could have a total trace that is greater than one. Note that equation 3.18 is split up to allow for the algorithm to be more efficient.

Algorithm 3.1 TD(λ)

Input:
S: The set of states in the world.
V: The tabular state value function.
π: The tabular state value function.
λ: The tabular state value function.
γ: The tabular state value function.

1: Initialise $V(s)$ arbitrarily for all $s \in S$
2: **Repeat** (for each agent):
3: $e(s) \leftarrow 0$ for all $s \in S$
4: Initialise s
5: **Repeat** (for each step of the agent's lifetime):
6: $a \leftarrow$ action given by π for s
7: Take action a, observe reward r and next state s'
8: $\delta \leftarrow r + \gamma V(s') - V(s)$
9: $e(s) \leftarrow e(s) + 1$
10: **Repeat** (for each $s'' \in S$):
11: $V(s'') \leftarrow V(s'') - \alpha\delta e(s'')$
12: $e(s'') \leftarrow \gamma\lambda e(s'')$
13: $s \leftarrow s'$

The new symbol ($\lambda \geq 0$) is a parameter of the algorithm that controls the weighting of how far the back-propagation of rewards goes. If λ is set to zero, then the reward is only ever propagated backwards by one step each time a state is visited. If λ is set to one, then the reward is propagated backwards through the entire chain of states visited with no weighting other than the time discounting. Values of λ in between zero and one progressively weight more towards the recent past as the value approaches zero. Note that λ is set to one, the algorithm produces the same output as a variant of the Monte Carlo approach.

Temporal-difference methods originally only focused on the state value function, and the introduction of the TD(λ) algorithm by Sutton (1988) made no mention of the state-action value function. The work that applied temporal

[3]Some minor changes were made for notational clarity, to comply with the vernacular used in this thesis and to take into account the errata by Sutton (2010).

difference methods to the state-action value function was done by Watkins (1989; Watkins & Dayan 1992). Watkins introduced the Q-learning algorithm, which is not quite the state-action version of the TD(λ) algorithm due to the use of off-policy learning. Off-policy learning is where the algorithm learns the optimal policy, but does not follow it. Instead an off-policy algorithm follows a separate, related policy that still allows for exploration. By taking into account the differences between the policy being followed and the optimal policy, the optimal policy can be updated such that it includes no exploration and so can truly be the optimal policy. There does exist an on-policy temporal difference method, which is a direct state-action version of the TD(λ) algorithm. It is known as Sarsa and was proposed by Rummery & Niranjan (1994).

3.3.7 Extensions to Temporal Difference Learning

Up to this point, this section has been discussing the core work of reinforcement learning. There are a great many extensions to the core, too many to cover in any detail. This subsection will look at two extensions to give an impression of how the reinforcement learning problem can be extended.

The aim of generalisation is to apply the knowledge that has been learned from the environmental states that have been observed to those states that have not been visited. This is done by adding to each state some supplementary information which the agent can use to compare the similarities between states. It is the job of the agent to learn which pieces of the supplementary information are useful when and by how much. The agent would then use the information to predict the value function of a state based on supplementary information alone.

How generalisation is done is the subject of a great deal of research. The general idea though is to create a state-value function that is parameterised by the values of the supplementary information rather than the individual state. However, in allowing for generalisation, there has to be the trade-off that no state or state-action value will ever be able to be perfectly predicted. This is because by using a general measure of similarity between states, means that the transition between states has to be smooth. Therefore, if two close states are near enough but give very different values for their value function, the smoothness of the general value function will mean both values are pulled away from their true value.

Because of this, the goal of a general value function is not to match each state exactly, but to minimise the mean of the squared error between the values of the true (optimal) value function and the general value function. This is done by a method known as gradient descent. The intuition here is that as a given value is changed, this will change the mean of the squared error in some manner, either up or down, meaning that there is a local gradient for each parameter value. By calculating this gradient for each parameter, the direction that most quickly reduces the mean of the squared error can be followed,

allowing for a minimum value to be found. Alonso *et al.* (2006) suggested an approach to generalisation for Q-learning based on a number of observations made regarding the differences between the TD learning method and the Rescorla-Wagner model of classical conditioning. Sutton *et al.* (2009) have produced a variant of TD learning that calculates gradient descent efficiently and in a convergent manner.

A related, but different extension to reinforcement learning is that of introducing the notion of continuous states and actions. In the conventional version of reinforcement learning, each state and action is a discrete set of choices. However this need not be the case, for instance if a robot is to choose an angle with which to turn through then this is an action where there is a continuum of possible actions rather than a discrete set of choices. Similarly the state that the robot is in after making its choices is also a continuum. The state space becomes an even larger continuum if the robot then moves forwards after the turn, with another continuum of choices. Some of the methods of generalisation can be applicable, as one form of representing this is to have the supplementary state information become the state itself. Work by van Hasselt (2012) has looked at this version of the reinforcement learning problem in detail. Some recent research by Fairbank & Alonso (2011; 2012) has looked at this variant in terms of dynamic programming.

3.3.8 Reinforcement Learning in Relation to this Thesis

This thesis has many commonalities with reinforcement learning in general and temporal difference learning specifically. It also has a number of very significant differences. This subsection will review these similarities and differences.

The similarities between reinforcement learning and temporal difference learning and this thesis are due to their common origins: conditioning. This means that both systems learn to associate temporally congruous events, as that is the basis for conditioning. Both temporal difference learning and the system presented in this thesis use a value of association to represent the learning – in temporal difference, this is the association to the reward (i.e. the value function) and in this thesis it is the significance value (see chapter four for details). There are other similarities that are not necessarily conditioning based, but are logical developments of it. These similarities are that both systems learn continuously, are able to adapt the circumstances over time and the converged learning need not be biased by the earliest values learned.

However, there are numerous differences, which again originate in decisions made regarding the interpretation of the phenomena of classical conditioning. The largest difference is that the system developed in this thesis does not make use of any concept of rewards, nor of action. This was due to subscribing to the stimulus-stimulus interpretation of conditioning. In the temporal difference system, the only thing learned is the propagation of reward. This means that if the environment provides no reward, no learning occurs. This is in-line with

the stimulus-response interpretation of classical conditioning, which in the current psychological consensus is that it is the minor form of association. It is accepted that in the reinforcement learning variant of temporal difference, that conformance to any psychological interpretation was never the goal, however given the models heritage, such analysis is to be expected.

Another point regarding the difference between the system presented by this thesis and reinforcement learning in general is that during the transition between the model of classical conditioning and the machine learning method, the basis of what is being learned subtly changed. No longer was the learning method one of classical conditioning but one of instrumental conditioning[4]. The nature of classical conditioning is that the subject learns passively, with no choice of actions - the unconditioned and conditioned responses are reflexive, not deliberate actions, and if the stimulus-stimulus interpretation is correct, the response is not needed for learning to occur[5].

In instrumental conditioning, the choices are deliberate actions on the part of the subject seeking the expectation of reward or avoidance of punishment. The nature of what becomes associated in instrumental conditioning is different from classical conditioning. In instrumental conditioning all three items of the cue stimulus, the action and the reinforcement stimulus are associated, whereas in classical conditioning there is no action, just a cue and a reinforcement stimulus being associated. Note that in reinforcement learning, it is indeed a triple of the current state, the action and the reward. A possible argument against this assertion that reinforcement learning is more analogous to instrumental conditioning is that one can argue that classical conditioning is to instrumental conditioning what the state value function is to the state-action value function. The counterargument is that just because the state value function does not select actions, the policy which it evaluates does select actions, meaning that action selection is still very much an essential part of reinforcement learning, even in the case of the state value function.

The consequence of this subtle difference between the classical conditioning origins of reinforcement learning and its instrumental conditioning abstraction means that the basis for temporal difference learning has unanswered questions. This raises the question that, if a temporal difference model of instrumental conditioning were formulated, would it be different to the classical conditioning model? If so, would that model be able to be applied to the reinforcement learning problem, potentially giving a better solution to it?

---

[4]Instrumental conditioning is a form of associative learning where the subject is given a cue stimulus, and the reward is based on the active actions that the subject makes in response to the cue. Instrumental conditioning was first found by Thorndike (1898) and was greatly advanced by Skinner (1938; 1962). Unlike classical conditioning, the subject makes non-reflex actions.

[5]For the potential argument that in fear conditioning, actions can be taken that appear to be deliberate, a counterargument would be that the reflex response – the response that becomes conditioned – is the fear response itself, not any actions which follow the fear response.

As with the discussion over interpretation, it is accepted that it was not the intention of the machine reinforcement learning version of temporal difference to be compared, however again it does at least warrant discussion due to the heritage of the method.

It should be pointed out that the two previous arguments, "temporal difference follows a stimulus-response interpretation" and "temporal difference is instrumental conditioning", are not necessarily contradictory positions. Whatever the mechanism or interpretation of classical conditioning and instrumental conditioning, the similarities are greater than the differences between the two. This means that there is a likely shared mechanism between instrumental and classical conditioning and therefore, it is probable that the interpretation of classical conditioning also applies to instrumental conditioning.

There are other lesser differences between temporal difference learning and the system presented by this thesis. The first of these is that the choice of knowledge representation is different. Temporal difference typically encodes its knowledge in its value function and the system presented by this thesis uses predicate logic. The second difference is that temporal difference learning typically associates linearly, whereas the system presented by this thesis associates hierarchically, though it is acknowledged that there are extensions to temporal difference learning that do deal with hierarchical association.

3.4 Visual Event Sequence Learning

The research into visual event sequences is a part of the distinctive sub-field of artificial intelligence known as computer vision. As the system discussed within this thesis uses visual events as its data source, this section provides a brief review of the techniques used in computer vision to identify event sequences. This is followed by a brief review of visual object tracking techniques.

3.4.1 Event Recognition

Computer vision research into event detection and classification appears to be application-driven. These applications broadly fit into two groups of applications. These are the analysis of broadcast television and automated surveillance. There appears to be a single main difference in approach independent of application. There are those systems that manually and explicitly model the classes of event being searched for (Foresti *et al.*, 2004; Cui *et al.*, 2007; D'Orazio *et al.*, 2009; Cristani *et al.*, 2007) and those systems that do not, dynamically creating a model instead (Dee & Hogg, 2009; Piciarelli *et al.*, 2008). It also should be noted that the method used to detect events is very dependent on the type of application. This suggests that the approach of research being application-led has in this instance failed to produce any methods that are general enough to be used across applications.

Systems analysing broadcast television are mainly focused on news broadcasts, where the events being the story segments in the recording (Hoogs *et al.*,

2003; Xu & Chang, 2007) and sport broadcasts, where the events generally being dependant on the sport – for example goal events in soccer (D'Orazio *et al.*, 2009; Liu *et al.*, 2009). In the analysis of television broadcasts, the most common method of detecting events is through temporally dividing the video into individual shots, classifying each shot based on a cluster analysis and using typical sequences of shots to detect events. These methods are typically augmented with concurrent analysis of the audio and closed caption streams present in the broadcast.

Automated surveillance can be split into those systems attempting to characterise the behaviour of all detected subjects (Dee & Hogg, 2009; Cristani *et al.*, 2007) and those that attempt to search for anomalous events (Cui *et al.*, 2007; Foresti *et al.*, 2004; Piciarelli *et al.*, 2008). In the surveillance domain, the most common method to detect events is through analysis of the trajectories of the detected objects. These trajectories are calculated through the use of an object tracking system. To find anomalous trajectories, the system is trained on sets of trajectories that are considered normal (either through manual labelling or through clustering trajectories and using those clusters with the highest frequency of use). Anomalous or interesting trajectories are then classified as such if they are an outlier to the trained sets.

3.4.2 Object Tracking

While the development of this thesis did not directly implement a full tracking system, using simulated data instead, the system is designed such that the input data is supposed to be the output of a tracker or simulation thereof. In addition, the input data simulator modelled some of the kind of noise that a tracker produces in its output. As these two parts of this thesis assume knowledge of a tracking system, and for the sake of completeness, this very large area shall be briefly discussed. Yilmaz *et al.* (2006) provide a comprehensive review of tracking techniques, in which they split the tracking techniques into three different categories: point tracking, kernel tracking and silhouette tracking.

Point tracking is based on finding a correspondence between salient points in one frame and the same points in the next frame. Salient points are points in the scene that are either easy-to-find points on the objects present in the scene (for example, corners) and/or points that are invariant to particular transforms that may be applied to the image.

Kernel tracking is the category of methods where the objects to be tracked are represented in a simple manner, such as a bounding box or an ellipse. A correspondence between frames is then obtained for each simple shape. The search for the area of each frame that defines the shape can then be constrained by searching the neighbourhood of the shape in the previous frame. Kernel tracking is typically used with techniques that attempt to create foreground-background segmentation, such as between changing and static regions of pixels in the image.

Silhouette tracking is the name given to the set of methods that attempt to model the outline of each class of objects to be tracked. These models are typically a complex polygon or spline and have control points to allow the model to deform within set bounds. Tracking is then the creation of correspondences between models in consecutive frames. The position and shape of the previous frame is used to guide the search for the position and shape in the next.

In dos Santos *et al.* (2009), it is noted that an object forming part of a dynamic scene can display two types of motion: intrinsic and extrinsic. Intrinsic motion is that motion that changes the appearance of the object by the movement of constituent parts of the object (e.g. limbs on a body). Extrinsic motion is motion where the position of the object changes relative to other objects. From the point of view of tracking systems, extrinsic motion can be detected by either a kernel tracker or a silhouette tracker, whereas intrinsic motion can only be detected by a silhouette tracker.

The system discussed in chapter four of this thesis assumes the use of a kernel tracking system. While a silhouette tracker would provide the most information and allow for a more general way to describe events, it has a number of practical concerns. Firstly, a silhouette tracker would require models of the outlines of the objects expected to be tracked. This assumption of what objects would be present in a scene would mean that the system would be less adaptable to novel objects.

3.5 Chapter Conclusion

This chapter has reviewed the ideas that are related to this thesis, either through this thesis making use of those ideas or through the ideas having a common conceptual heritage. Firstly the work done within the psychology community towards modelling classical conditioning was reviewed. It was noted that there are two classes of model, trial-level models, and real-time models. A trial-level model is one that computes the association strength after each presentation of a stimulus completes. A real-time model computes the association strength at regular intervals.

Next, the ideas of commonsense knowledge were reviewed, with the three problems of knowledge representation, knowledge reasoning and knowledge acquisition. The two most pertinent problems, those of representation and acquisition were then discussed. The problem of knowledge representation focused on space and time. The problem of acquisition was reviewed more generally, looking at two dominant approaches of acquisition, namely using an interface that abstracts away the technical details of representing knowledge and that of using natural language processing to parse the commonsense knowledge that is expressed within large corpora such as web pages.

The chapter then went on to review reinforcement learning, looking at the reinforcement learning problem in detail, including its formulation as max-

imising a value function and as a Markov decision process. The three methods of solving the reinforcement learning problem were then briefly reviewed: dynamic programming, Monte Carlo and temporal difference techniques. The temporal difference techniques were then looked at in further detail due to their common conceptual heritage to this thesis. The similarities and differences between reinforcement learning and this thesis were then discussed, with the most notable difference being that this thesis does not assume external actions and rewards are available.

Finally, the chapter reviewed some of the literature surrounding visual event detection and a quick overview of tracking methods was included for completeness. The remainder of this thesis will be dedicated to describing and evaluating a system that makes use of the analysis presented in chapter two in a system that passively learns from classical conditioning. In particular, the next chapter presents a description of the system built to learn in the manner similar to classical conditioning.

Chapter 4

The System

In order to test the ideas expressed by the hypotheses presented in chapter one and expanded upon in chapter two, a system was developed that makes use of the ideas of classical conditioning to passively learn a model of the observed environment. This chapter describes that system.

Chapter two presented and analysed a wide range of phenomena of classical conditioning. It was not feasible within one project to implement all the features that were presented and analysed. Because of this constraint, the system implements a sub-set of the phenomena. The phenomena that were chosen to be implemented were those that were observed to be the most widely discussed within the classical conditioning literature. To recap, the phenomena that the system implements are:

1. Acquisition
2. Extinction
3. The Inter-Stimulus Interval
4. Reacquisition
5. Blocking
6. Recovery from Blocking
7. Conditioned Inhibition
8. Extinction of Conditioned Inhibition
9. Latent Inhibition
10. U.S.-Pre-Exposure Effect
11. Sensory Preconditioning
12. Secondary Conditioning

The system takes frame-based data and learns a model in an unsupervised manner with no external feedback. This is one of the main departures from previous reinforcement learning systems in that previous reinforcement learning systems require external evaluative feedback in the form of an external reward signal. The other main departure is that the system does not interact with the environment in the form of action selection. The basis for

these departures is that the stimulus-stimulus interpretation of classical conditioning implies there is no inherent need for reward and action to be present for learning to occur. Actions and rewards exist within classical conditioning experiments only to allow the learning process to be observed.

Some general terminology needs to be introduced that is used throughout this chapter. The term "event instance" refers to a specific observed occurrence, with a particular form, start time and end time. The term "event type" can then be defined to be the classification of the form of an event. The form of an event instance is that event instance's event type.

The chapter starts with a brief outline of how the system works overall, discussing only the key general ideas that are a part of its operation. The next four sections each provide an intuitive description of the function of the four modules that comprise the system. The final section of this chapter then provides a more formal description of the system and its implementation.

4.1 System Outline

In order to test the hypotheses, a system that creates a human-examinable environmental model using the principles of classical conditioning is required. This section gives an outline description of how the system works to produce this environmental model.

The model of the environment that is learned by the system consists of patterns of events that the system believes correspond to real-world phenomena. Due to the system learning patterns of events, the system requires a stream of events as input. In practical terms, this can limit the type of data that it will process. This is due to the practical need that those events from which the patterns are learned have to be defined. It is these event definitions that define the domain over which the pattern learning occurs.

The system presented in this chapter is provided with a sequence of frames, each of which is associated with a collection of object location and size data. These frames are then processed into a stream of pre-defined basic events based on the spatial and temporal relationships found for each object in each frame. The system then learns patterns of these events. This domain of the pre-defined events was chosen due to its flexibility in what could be learned and the ability for the learned patterns to be intuitively interpreted.

The system is implemented as four modules, representing the main processing steps of the system. These modules are: pre-processing, recognition, association and significance. The pre-processing module turns the object movement data into the pre-defined events from which the patterns are learned. The recognition module recognises existing event patterns and generates tokens representing those patterns. The association module associates the pattern tokens based on their timing to produce new candidate patterns. Finally, the significance module collects the evidence of new patterns and determines whether a given pattern is believed to exist based classical conditioning.

The system is based on a feedback loop between the pattern recognition module, the association module and the significance module. Figure 4.1 depicts the modules and the data that is passed between each module.

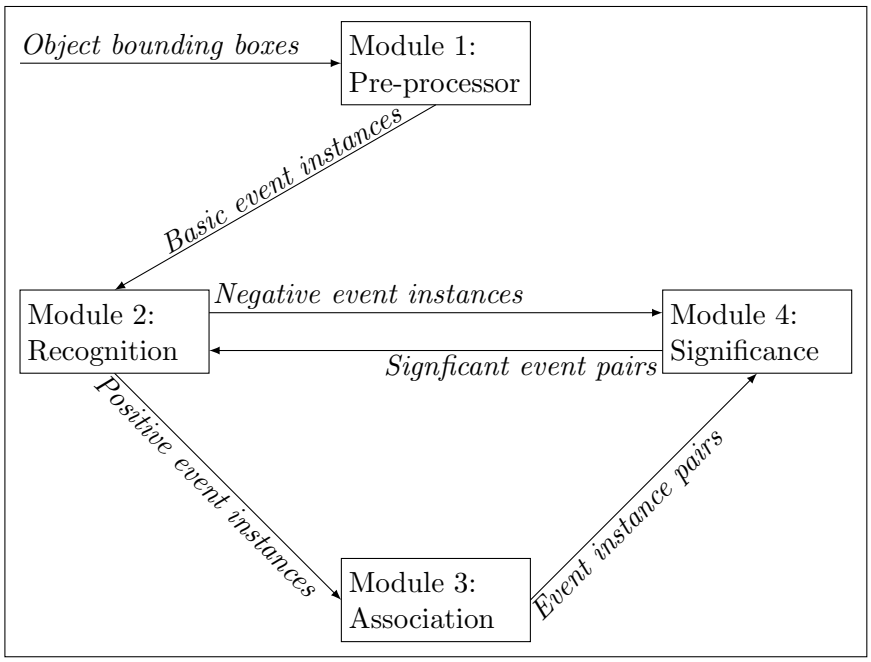

Figure 4.1: The four modules.

As input to the system, the system incrementally takes time-frames that comprise of bounding boxes for each object of interest in the observed scene. The first module takes these boxes and notes the spatial relationships between the objects. The module then recognises specific changes between each frame's spatial relationships and uses these changes as the basic event instances which the recognition system uses to recognise patterns of those events.

The second module recognises patterns of events. Each pattern is considered an event in its own right. An event pattern is a set of two time-ordered events, which can be either a basic event or another event pattern. The module takes each frame of basic events and compares those events with a list of known event patterns. When an event pattern is recognised, an event instance of that event pattern is generated for the current frame. When only the first event of an event pattern is recognised, but the second is not found, a different kind of event instance of that event type is generated for the current frame. The former kind of event instance is called a positive event instance; the latter kind of event instance is called a negative event instance. The second module then, in a recursive manner, uses the positive event instances as if they were also input events for the current frame to recognise higher-order patterns of patterns, again generating both positive and negative instances. The set of all positive event instances are passed to the third module and the set of all negative event instances are passed to the fourth module.

The third module, the association module, identifies pairs of those event instances whose temporal relationship satisfies a set of criteria such that they

can be said to happen together. The set of all event pairings is then passed to the fourth module.

The fourth module takes input from modules two and three. From the second module, it takes the negative event instances. From the third module, it takes the identified event pairings. To these inputs, the module applies various metrics that are grounded in the ideas of classical conditioning. The two inputs are treated as evidence for the existence or non-existence of the event-type pattern they represent. The identified event pairs are treated as positive evidence for the pattern they make together and the negative event instances are treated as negative evidence for their corresponding pattern.

This application of metrics in the fourth module results in a list of patterns together with a measure of how well the two parts of the pattern are associated, calculated by a measure based on some of the ideas of classical conditioning. The list is then made available to the second module where those patterns that have a measure value above a given threshold are used as the patterns that the second module recognises. Should the measure of an event pattern that was high enough to be recognised by the first module subsequently weaken such that it is no longer considered high enough, that event pattern is no longer recognised.

The definition of a recursive event type leads to further terminology that this thesis uses to describe the system. Firstly the terms "atomic event type" and "atomic event instance" respectively refer to the event types and instances that are the indivisible basic events that form the input to the system. Secondly the terms "composite event type" and "composite event instance" respectively refer to those event types and instances that are composed of other event types and instances. Finally the terms "component event type" and "component event instance" respectively refer to the event types and instances that make up a composite event type or instance.

4.2 Module 1 – Pre-Processor

The recognition module attempts to recognise patterns of event types. This requires some pre-processing of the system's input data to produce the atomic event instances from which the recognition module can use to recognise patterns, which is the job of the pre-processor module. This processing is largely the same as the atomic event type recognition presented by dos Santos *et al.* (2009), though with some differences.

There are two stages to the pre-processing, firstly the module computes a sequence of frame states from the system input and secondly the module recognises and generates instances of the atomic event types from each sequential pair of frame states. After describing the nature and format of the system input, this section will in turn look at the frame state computation stage followed by the atomic event type recognition stage.

4.2.1 System Input

The input to the system is a list of predicates, each predicate corresponding to a single frame of a video. Each predicate contains the coordinates and sizes of a set of boxes that describe the boundary of an object or objects of interest that appear within that frame. This format is the same as the format described by Bennett *et al.* (2008, p. 72). Figure 4.2 is an annotated example of the data format used.

Not all the data available is made use of. This format was chosen to be backwards-compatible with the system described by Bennett *et al.* (2008). This system only makes use of the frame number, object labels, box position and minimum box size data. Effectively, the measures of uncertainty within the tracking data is currently ignored by this system, for two reasons, firstly it allows for a simpler design and secondly it is believed that the mechanisms the system has to deal with noise should compensate for uncertainty in the input data.

4.2.2 Frame State Calculation

The first stage of pre-processing is to extract the relevant qualitative states of each frame of the input. Each individual frame of the system input is iteratively processed to find a set of variables that describe the state of the objects of that frame and the state of the relationships between each pair of objects. These variables are similar to the variables that are calculated by dos Santos *et al.* (2009); differences will be noted as each variable is discussed. There are three per-object variables and four variables that describe the relationships between each pair of objects. The first per-object variable, a variable not used by dos Santos *et al.* (2009), marks whether an object is currently visible[1]. The other two per-object variables are the x and y positions of each object (if more than one object is a part of the same box, then both objects share the same position). The four variables that describe the relationship between each pair of objects are the straight-line distance between the two centres of both objects, the connectivity between the two objects, the horizontal relationship between the two objects and the vertical relationship between the two objects.

The connectivity of the two objects represents whether the objects appear to be touching one another and if so, how. This is encoded by means of three fluents which are mutually exclusive. Table 4.1 describes the meaning of each predicate. The fluents are the the RCC-3 variant of the Region Connection Calculus (Randell *et al.*, 1992) by Santos & Shanahan (2002). The original Region Connection Calculus has eight different states, but these assume that the relative positions of each region can be perfectly distinguished, even when one object is completely enclosed within the other. The variant by Santos & Shanahan (2002) assumes that regions cannot be perfectly distinguished.

[1]For the purposes of scalability, the system assumes that by default a predicate is false if it has not been asserted.

```
frame( 9, 2,                                  Frame number, box count
       [                                              List of boxes
         box( 9, 18, 16,          Frame number, box ID, parent box ID
              [person],                 List of detected object labels
              [0.89],                List of object existence probabilities
              [                                   List of box geometry
                [291.5, 352],          Coordinates of the box centre
                [93, 308],                  Minimum box size (x, y)
                [100, 314]                  Maximum box size (x, y)
              ]),
         box( 9, 19, 17,
              [ball],
              [0.99],
              [
                [285, 197],
                [46, 46],
                [52, 53]
              ])
       ]).

frame( 10, 1,
       [
         box( 10, 20, 18,
              [person, ball],
              [0.81, 0.85],
              [
                [293, 340.5],
                [96, 331],
                [105, 340]
              ])
       ]).
```

Figure 4.2: An annotated example of the system's input data.

Fluent	**Meaning**
co (o_1, o_2)	o_1 *is coalescent with* o_2: The boxes of the two objects o_1 and o_2 are either the same or overlap to the extent that the two objects cannot be reliably distinguished.
extC (o_1, o_2)	o_1 *is externally connected with* o_2: The boxes of the two objects o_1 and o_2 are touching but do not overlap greater than a given error margin.
disC (o_1, o_2)	o_1 *is disconnected with* o_2: The boxes of the two objects o_1 and o_2 are distinctly separate.

Table 4.1: A list of the meanings of the connectedness predicates available.

Because the Santos & Shanahan (2002) Region Connection Calculus variant assumes that regions cannot be perfectly distinguished, it implies that the boundary of a region is also uncertain. This assumption also implies that as two objects transition between one of the states and another, there is some uncertainty as to when the transition happens. The way the states are calculated takes both of these factors into account.

The coalescence fluent holds either when the two objects are contained in the same box or when there is an area of overlap between the two boxes that as a percentage of the smallest box area is above a given threshold (τ_{area}). This criterion is expressed in equation 4.1. The external connection fluent holds when there is an overlap below the threshold τ_{area} and when there is no overlap and the distance between the nearest two points on the box boundaries (calculated by the function $\text{dist}(o_1, o_2)$) is below a given threshold (τ_{dist}). This criterion is expressed in equation 4.2. The disconnection fluent holds when the distance between the nearest two points on the box boundaries is above τ_{dist} This criterion is expressed in equation 4.3.

$$\texttt{co}\,(o_1, o_2) \leftrightarrow \tau_{\text{area}} \leq \frac{\text{area}(o_1 \cap o_2)}{\min\left(\text{area}\,(o_1)\,, \text{area}\,(o_2)\right)} \tag{4.1}$$

$$\texttt{extC}\,(o_1, o_2) \leftrightarrow \tau_{\text{area}} > \frac{\text{area}(o_1 \cap o_2)}{\min\left(\text{area}\,(o_1)\,, \text{area}\,(o_2)\right)} \wedge \tau_{\text{dist}} > \text{dist}\,(o_1, o_2) \tag{4.2}$$

$$\texttt{disC}\,(o_1, o_2) \leftrightarrow \tau_{\text{dist}} \leq \text{dist}\,(o_1, o_2) \tag{4.3}$$

These criteria differ in some aspects from Santos & Shanahan (2002) and therefore also differ from dos Santos *et al.* (2009), as that paper uses the same criteria. The way they differ is that Santos & Shanahan define the criterion for coalescence to be when the distance between nearest two points on the box boundaries is zero – the distance threshold (τ_{dist}) separating the states external connection and disconnection remains the same. By using the overlap area percentage, it takes into account the sizes of the objects involved; an overlap area of 10 pixels may be considered small for an object with a total area of 1000 pixels but be considered large for an object with a total area of 20 pixels. This is not taken into account with the criterion used by Santos & Shanahan (2002), which assumes any overlap should be considered to be a sign of coalescence, which does not respect the original implication that there needs to be a margin of tolerance in the region boundaries.

The horizontal and vertical relationships between each pair of objects refer to the positioning in relation to each other. Each relationship is separately encoded by means of three mutually exclusive fluents: $\texttt{left}\,(o_1, o_2)$, $\texttt{inlineX}\,(o_1, o_2)$ or $\texttt{left}\,(o_2, o_1)$ for the horizontal relationship and $\texttt{above}\,(o_1, o_2)$, $\texttt{inlineY}\,(o_1, o_2)$ or $\texttt{above}\,(o_2, o_1)$ for the vertical relationship. Table 4.2 describes the meaning of each predicate. The first three fluents refer to the horizontal relationship and the last three refer to the vertical relationship. These fluents can be seen to be similar to those proposed by Frank (1992), but instead of defining 9 states of the compass, the relations are

separated into their horizontal and vertical components and then only one of the two non-inline relations is defined, relying on the other being defined by transposing the object symbols. This extends the expressiveness of the positioning fluents used by dos Santos *et al.* (2009), which only defined the `left` fluent. These predicates are all calculated using the position of the centre of one box in relation to the border of the other box. For example, $\texttt{left}\,(o_1, o_2)$ holds when the centre of the box of object o_1 is to the left of the left box edge of object o_2; $\texttt{inlineX}\,(o_1, o_2)$ holds when the centre of the box of object o_2 is between the left and right edges of the box of object o_1 and $\texttt{left}\,(o_2, o_1)$ holds when the centre of the box of object o_2 is to the right of the left box edge of object o_1.

Fluent	**Meaning**
$\texttt{left}\,(o_1, o_2)$	o_1 is to the left of o_2
$\texttt{left}\,(o_2, o_1)$	o_2 is to the left of o_1
$\texttt{inlineX}\,(o_1, o_2)$	Both o_1 and o_2 are in-line in the x axis Note that $\texttt{inlineX}\,(o_1, o_2) = \texttt{inlineX}\,(o_2, o_1)$
$\texttt{above}\,(o_1, o_2)$	o_1 is above o_2
$\texttt{above}\,(o_2, o_1)$	o_2 is above o_1
$\texttt{inlineY}\,(o_1, o_2)$	Both o_1 and o_2 are in-line in the y axis Note that $\texttt{inlineY}\,(o_1, o_2) = \texttt{inlineY}\,(o_2, o_1)$

Table 4.2: A list of the meanings of the horizontal and vertical object inter-relation predicates available.

Equations 4.4 to 4.9 provide definition for all of the horizontal and vertical object relationship fluents. The function $\text{pos}_x\,()$ returns the horizontal position of the centre of the object and similarly, the function $\text{pos}_y\,()$ returns the vertical position of the centre of the object.

$$\texttt{left}\,(o_1, o_2) \leftrightarrow \text{pos}_x\,(o_2) > \text{pos}_x\,(o_1) + \frac{\texttt{width}\,(o_1)}{2} \tag{4.4}$$

$$\texttt{left}\,(o_2, o_1) \leftrightarrow \text{pos}_x\,(o_2) < \text{pos}_x\,(o_1) - \frac{\texttt{width}\,(o_1)}{2} \tag{4.5}$$

$$\texttt{inlineX}\,(o_1, o_2) \leftrightarrow \neg\texttt{left}\,(o_1, o_2) \wedge \neg\texttt{left}\,(o_2, o_1) \tag{4.6}$$

$$\texttt{above}\,(o_1, o_2) \leftrightarrow \text{pos}_y\,(o_2) > \text{pos}_y\,(o_1) + \frac{\texttt{height}\,(o_1)}{2} \tag{4.7}$$

$$\texttt{above}\,(o_2, o_1) \leftrightarrow \text{pos}_y\,(o_2) < \text{pos}_y\,(o_1) - \frac{\texttt{height}\,(o_1)}{2} \tag{4.8}$$

$$\texttt{inlineY}\,(o_1, o_2) \leftrightarrow \neg\texttt{above}\,(o_1, o_2) \wedge \neg\texttt{above}\,(o_2, o_1) \tag{4.9}$$

To summarise this subsection, table 4.3 lists each of the variables that are calculated to embody the state of the input frame. For each variable, it lists the variable, the possible fluents that can represent the state, and any constraints for the fluent. Note that in the list of constraints the set *Objects* refers to the set of all objects recognised by the system (independent of whether an object is visible at any particular time), the set $\mathbb{N}_0$ refers to the set of natural

numbers (including zero) that can be represented by the computer and the set $\mathbb{R}_0^+$ refers to the set of non-negative real numbers that can be represented by the computer.

Variable	Fluent	Constraints
Object o is visible in the scene.	$\texttt{visible}\,(o)$	$o \in Objects$
Object's x position	$\text{pos}_x\,(o, x)$	$o \in Objects$, $x \in \mathbb{N}_0$
Object's y position	$\text{pos}_y\,(o, y)$	$o \in Objects$, $y \in \mathbb{N}_0$
Distance between objects o_1 and o_2	$\text{dist}\,(o_1, o_2, d)$	$o \in Objects$, $d \in \mathbb{R}_0^+$
Connectedness relationship between objects o_1 and o_2	$\texttt{co}\,(o_1, o_2)$ $\texttt{extC}\,(o_1, o_2)$ $\texttt{disC}\,(o_1, o_2)$	$o_1, o_2 \in Objects$, $\texttt{co}\,(o_1, o_2) = \texttt{co}\,(o_2, o_1)$ $\texttt{extC}\,(o_1, o_2) = \texttt{extC}\,(o_2, o_1)$ $\texttt{disC}\,(o_1, o_2) = \texttt{disC}\,(o_2, o_1)$
Horizontal relationship between objects o_1 and o_2	$\texttt{left}\,(o_1, o_2)$ $\texttt{left}\,(o_2, o_1)$ $\texttt{inlineX}\,(o_1, o_2)$	$o_1, o_2 \in Objects$ $\texttt{inlineX}\,(o_1, o_2) = \texttt{inlineX}\,(o_2, o_1)$
Vertical relationship between objects o_1 and o_2	$\texttt{above}\,(o_1, o_2)$ $\texttt{above}\,(o_2, o_1)$ $\texttt{inlineY}\,(o_1, o_2)$	$o_1, o_2 \in Objects$ $\texttt{inlineY}\,(o_1, o_2) = \texttt{inlineY}\,(o_2, o_1)$

Table 4.3: A list of the variables used to represent the state of a frame of system input.

4.2.3 Atomic Event Calculation

The atomic event types denote the change of a state between two consecutive frames. Table 4.4 lists the event type fluents and their meaning. The definition of each event type is based upon the frame-state variables and is listed in Appendix A. There are no constraints as to the mutual exclusivity or otherwise of these event types past the event type definitions themselves. This implies that it is possible for an object to move left and up at the same time, but can't move left and right at the same time due to the atomic event type definitions making this a mathematical impossibility.

The processing takes place using sequential pairs of frame-states. One frame-state is designated as the current frame-state and the other as the previous frame-state. In this way, the current frame-state becomes the previous frame-state when the next frame-state is processed. The pairs of frame-states are compared against the atomic event type definitions and the atomic event types that match the frame-states are generated as instances and are passed to the recognition module.

The event types that are detected are based on those described by dos Santos *et al.* (2009) but have a number of significant extensions. The remain-

Fluent	**Meaning**
`lost` (o)	Object o is no longer visible.
`found` (o)	Object o has become visible.
`moveLeft` (o)	Object o has moved left.
`moveRight` (o)	Object o has moved right.
`moveUp` (o)	Object o has moved up.
`moveDown` (o)	Object o has moved down.
`approaching` (o_1, o_2)	Object o_1 and Object o_2 approached each other.
`receding` (o_1, o_2)	Object o_1 and Object o_2 receded from each other.
`mergeRight` (o_1, o_2)	Object o_1's track box has merged with Object o_2's track box on the right of Object o_2
`mergeLeft` (o_1, o_2)	Object o_1's track box has merged with Object o_2's track box on the left of Object o_2
`mergeTop` (o_1, o_2)	Object o_1's track box has merged with Object o_2's track box on the top of Object o_2
`mergeBottom` (o_1, o_2)	Object o_1's track box has merged with Object o_2's track box on the bottom of Object o_2
`emergeRight` (o_1, o_2)	Object o_1's track box has emerged from Object o_2's track box on the right of Object o_2
`emergeLeft` (o_1, o_2)	Object o_1's track box has emerged from Object o_2's track box on the left of Object o_2
`emergeTop` (o_1, o_2)	Object o_1's track box has emerged from Object o_2's track box on the top of Object o_2
`emergeBottom` (o_1, o_2)	Object o_1's track box has emerged from Object o_2's track box on the bottom of Object o_2
`makeContactRight` (o_1, o_2)	Object o_1 has made contact with Object o_2 on the right of Object o_2
`makeContactLeft` (o_1, o_2)	Object o_1 has made contact with Object o_2 on the left of Object o_2
`makeContactTop` (o_1, o_2)	Object o_1 has made contact with Object o_2 on the top of Object o_2
`makeContactBottom` (o_1, o_2)	Object o_1 has made contact with Object o_2 on the bottom of Object o_2
`breakContactRight` (o_1, o_2)	Object o_1 has broken contact with Object o_2 on the right of Object o_2
`breakContactLeft` (o_1, o_2)	Object o_1 has broken contact with Object o_2 on the left of Object o_2
`breakContactTop` (o_1, o_2)	Object o_1 has broken contact with Object o_2 on the top of Object o_2
`breakContactBottom` (o_1, o_2)	Object o_1 has broken contact with Object o_2 on the bottom of Object o_2

Table 4.4: A list of the fluents used to represent atomic event types along with their meaning.

der of this section will look at these differences. The first difference is that dos Santos *et al.* (2009) did not directly record the time of an event instance or frame-state; instead a new connective was defined, called serial conjunction. The serial conjunction connective, $\otimes$ connects two predicate logic statements ρ_1 and ρ_2 such that the resultant statement $\rho_2 \otimes \rho_2$ means "ρ_1 happens immediately before ρ_2".

In contrast the system described in this chapter uses a more general relationship than serial conjunction, in that it allows for serial, parallel and overlapping conjunction. This is achieved by pairing events hierarchically, and will be described in section 4.3.

The remaining changes between the two systems are additions or subtractions of atomic event types. These are listed below:

- dos Santos *et al.* (2009) only defined event types along the horizontal axis. The system described by this thesis defines the relevant event types in terms of both the horizontal and vertical axes.

- Event types were added that represent objects entering and leaving the observed scene. These additions were influenced by Ivanov & Bobick (2000).

- The system also has an addition of event types to represent the absolute movement of objects. These were included as it was deemed that only using relative movement (as was done by dos Santos *et al.* (2009)) is insufficient – for instance, the event type approaching can be satisfied both by a mobile object moving towards a static object and a chase scenario where the chasing object is gaining on the chased object. It may be desirable to allow for these situations to be differentiable and this can be done through representing absolute movement. These issues are discussed further by Van de Weghe *et al.* (2005).

- Event types that represent objects making and breaking contact were added. The merge and emerge event types are transitions to and from a coalescent state. The addition of event types representing objects making and breaking contact allow for the transitions to and from an externally connected state. The definitions of these event types recognise that for two objects to transition between being disconnected to being coalescent, the objects need to pass through an externally connected state and so, the make or break contact event types are generated even if the transition between the disconnected and coalescent states occurs over less than one frame.

- dos Santos *et al.* (2009) provided an event type to represent a static relationship between two objects. This has been removed as it represents no change of state.

- dos Santos *et al.* (2009) also provided a set of three event types to represent an object's relationship to the camera (approaching, receding and

stasis). These were removed as they were calculated based upon the change in the size of an object. This makes the assumption that an object's absolute size is constant, which is not necessarily true. These event types could only be correctly defined in a system that employs stereo cameras (even in this case, the event types may be better implemented by introducing an observer object into the scene and using the existing approaching and receding event types).

4.3 Module 2 – Recognition

The recognition module turns an input stream of atomic event instances that are all one frame long into two streams of instances of recognised patterns of event types. These instances can be of different frame lengths. The first stream consists of event type patterns where all constituent events happened. The second stream consists of event type patterns where only some of the constituent event types happened within the time allowed.

This section is divided up as follows. First the core concept of the recognition system is discussed, namely hierarchical events. Following-on from this, the complicating factors are discussed, such as event instances that happen over more than one frame. The next three subsections extend the core concept in different ways to make allowances for the complicating factors. Those three subsections are followed by an explanation of how the extending concepts work together to allow for the complicating factors. The final subsection discusses how the system as a whole creates the two output streams.

4.3.1 Hierarchical Events

The system considers that patterns of event types are event types in their own right. This means that the system has a recursive hierarchy of event types where patterns of patterns and patterns of patterns of patterns and so on occur. These patterns of event types are called compound event types, in contrast to the atomic event types. Figure 4.3 demonstrates a three-level hierarchy of event types. By creating a hierarchy of patterns, sub-patterns can be re-used within other patterns. This re-use is found in figure 4.3 in event types A, B and C on the first level and event type 2 on the second level.

Taking this patterns-are-event-types concept to its limit, any pattern of event types can be represented as a hierarchy of compound event types where each compound event type is composed of only pairs of event types. For example, take event type 1 of figure 4.3, with its pattern ABAC. By taking each event type in turn and pairing it with its next event type, the pattern can be decomposed into the event type pairs (A,B), (B,A) and (A,C). These event type pairs can then again be sequentially composed into the pairs ((A,B),(B,A)) and ((B,A),(A,C)) which in turn can be finally composed into the pair (((A,B),(B,A)), ((B,A),(A,C))). This decomposition is represented graphically in figure 4.4.

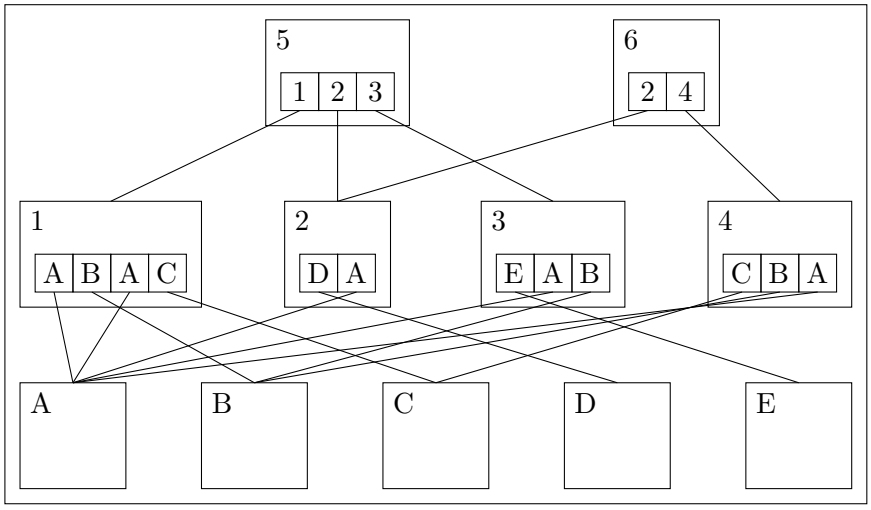

Figure 4.3: A 3-level hierarchy of event types. Letters correspond to atomic event types (the first level) and numbers correspond to composite event types (the second and third levels).

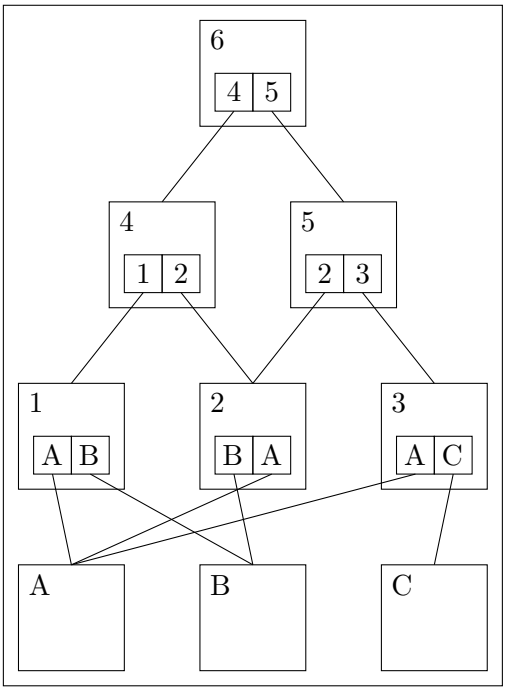

Figure 4.4: A hierarchy of event types involving only two-component compound event types.

To match patterns, each composite event type can be said to be attempting to search for its pattern within the stream of atomic event instances and when it is successful, inserts an instance of its event type into the event instance stream. When an event type does insert an event instance into the event instance stream it allows for event types higher up in the hierarchy to match their pattern.

For an example of this matching behaviour the pattern of figure 4.4 ('ABAC') shall be used. Consider the situation where the event types A and B have been observed over the last two frames. Due to this observed pattern, event type 1 has been matched and an instance generated. Therefore the most recent frame has the event instance set {B, 1} which causes event types 2 and 4 to both be expecting that their next event type will occur soon. If the next frame event instance set contains an instance of event type A, then the resultant event instance set for the frame will be {A, 2, 4} leading to event types 3, 5 and 6 to all expect that their next event type will occur soon.

The reason it is useful to create such a hierarchy of event type pairs is that it allows for the question of learning arbitrary event type patterns to be phrased as a matter of learning what event type is to be expected given an event instance. The task of the system is to learn which pairs of event types the system encounters are significant enough to deem that they are a pair of event types that forms a pattern and not noise. This is where the concepts of classical conditioning are primarily used, as will be discussed in section 4.5. The significance module is responsible for determining the significance of pairs of event types.

The cost of using hierarchical patterns of event types is that it increases the space required to store a particular pattern. For a pattern of length n, the worst case number of event types required to represent the pattern is $\frac{n(n+1)}{2}$ (i.e. the space requirements are $O\left(n^2\right)$). The worst case occurs when there is no repetition of any event type. If there is any repetition, then significant space savings are made. For instance, each re-use of an atomic event type effectively reduces the pattern length by 1, giving a space saving for the i^{th} re-use of $n-(i-1)$ and a total space saving of $\sum_{i=1}^{m}(n-(i-1)) = m\left(n-\frac{m-1}{2}\right)$ where m is the total number of re-uses.

As mentioned briefly, the responsibility for asserting the existence of event type pairings is undertaken by the significance module. This is done by processing the evidence for the existence of a pairing and assigning a significance measure or measures. The significance module then makes available the asserted event type pairings to the recognition module to be used as the event type patterns to search for.

4.3.2 Multi-Frame Events

The previous subsection described basics of how the system recognises event type patterns. However, that discussion only looked at event instances that happen over a single frame, to allow the core principles to be discussed free from complicating factors. This and the next four subsections look at extending that description to account for event instances that happen over more than one frame. This includes three cases – on-going event instances, delayed event instances and overlapping event instances.

On-going event instances are atomic event instances that are detected over multiple consecutive frames but each frame instance is in reality just multiple detections of a single long event instance. The `approaching` event type is an example of such an event type capable of the behaviour – it can be detected over a single frame but when detected in two consecutive frames, those two detections are in reality part of a single long event instance.

Delayed event instances refer to composite event instances where there are one or more frames between the first component event instance and the second component event instance. In the intervening time, other event instances that are not a part of that pattern could take place.

Due to allowing for multiple atomic event instances to happen in any one frame, multiple patterns may be detected simultaneously. Combined with the existence of on-going event instances and delayed event instances, it is possible for two event instances to overlap. Some higher-order event type patterns may involve pairs of composite event types that can either always overlap or occasionally overlap. In order to detect these types of overlapping patterns, the system needs to allow for the hierarchy to detect overlapping patterns; this is referred to as overlapping event types.

The way that these cases can be dealt with is the topic of the next four subsections, the first three each introducing an extending concept to the recognition system presented so far. The first of these is growing event instances, the second is the generation of a type of event instance known as a potential event instance and the third concept is the use of a moving window. The final subsection shows how these new extensions handle these three cases of event instances that exist over multiple frames.

4.3.3 Event Growing

Event growing refers to the reconstruction of on-going event instances. This is achieved by examining the event instances generated in the current frame and event instances generated in the previous frame. Where an instance of the same event type exists in each frame, the event instance of the previous frame is removed and the start time of the current event instance is changed to the start time of the previous event instance. This way, event instances (particularly atomic event instances) that occur over multiple frames are extended as they are observed to occur and cease to be extended when they don't. This allows

for the construction of on-going event instances, and as will be shown later, the interaction of event growing with the other two extensions allows for the three cases of event type patterns to be incorporated within the hierarchical recognition system.

For atomic events, event growing only happens over consecutive frames, and not over any further gap. If an event instance was grown where there is a delay, then it could allow for mutually exclusive event types to happen simultaneously – a contradiction. For example, consider the case where two objects first approach, then recede and then approach again. If the event type `approaching` was merged into one long event instance, then for the middle frame, the system would record the two objects as having approached each other and receded from each other at the same time.

4.3.4 Potential Event Instances

When an atomic event instance occurs that is a part of a single composite event type, it creates multiple expectations at once. It both creates an expectation that the second component atomic event type will occur and therefore also an expectation that the composite event type itself will occur. If that composite event type is in turn a part of a higher-level composite event type, then there is also some expectation that the higher-level composite event type will too come to pass. This chaining of expectations can carry-on up the entire structure of the event type hierarchy to the highest level relevant composite event type. In relation to the example of figure 4.4, if event type A was encountered with no prior expectations, hierarchically there would be an expectation of event types 1, and 3, and an expectation of event type 1 would create an expectation for event type 4 which in turn would create an expectation for event type 6. Event type 3 does not create any further expectations because it is not used as the first event type of any higher-level event types.

What the system does is, when any atomic event type is observed, an event instance is generated for every composite event type where that atomic event type occurs as the first component. Each of those instances is marked as being a potential event instance. This generation of potential event instances is repeated for every applicable layer of the hierarchy. When the second component event type of a composite event type is observed, the potential event instance is replaced with a non-potential event instance of the same type that starts at the same time as the potential event instance and ends at the same time as the second component event instance (which will always be the same as the current time as the second event instance will have just been discovered). These potential event instances are how the state of expectant composite event types are stored.

The replacement of event instances due to event growing and the replacement of event instances due to potential event instances have a different basis. The event instance replacement of event growing occurs due to the detection of event instances of the same type whereas the event instance replacement

of potential event type confirmation occurs due to the detection of event instances of differing types. The practical consequence of this difference is that while the former needs to be restricted to only replacing subsequent event instances, the latter can be allowed to replace event instances where there is a gap between the two detected parts. This fact is utilised by the concept of a moving event window, which is discussed next.

4.3.5 The Moving Window

In chapter two, it was argued that the loss of association strength as the inter-stimulus interval increases implies that there is a cut-off point for two event instances to be associated at all and that this is a reasonable strategy in learning to predict one event type given an instance of another. This need for a finite cut-off point can be used by this system. Because computers are by their nature discrete, all asymptotic functions reach their asymptote in a finite number of steps, due to the difference between the asymptote and the function reducing below the smallest number representable by a finite memory computer. Therefore, if the function that determines the change in association strength asymptotically reduces to zero as the time between the two event instances increases, there exists a point where the time between the two event instances is so large that it causes no change to the association strength. This means that there naturally exists a moving window where it is impossible to learn any association between two delayed event instances. This moving window is of a fixed length and always finishes at the current frame.

In similar vein, there also exists a smaller moving window where there is a change in the association strength but that that change is so small to not be worth the cost of keeping track of the event instances involved to apply changes over longer periods of time. Depending on what threshold is set for what is considered too small a change, the corresponding window size may be many orders of magnitude smaller than the natural window described previously.

The existence of the moving window means there is a cut-off point after which there is no need to recognise a delayed event instance, because the composite event type will never be learned by the system in the first place. Therefore, the recognition system only needs to keep track of the set of event instances where the end frame number of each event instance is a fixed number of frames in the past.

The system keeps track of event instances within the window through the use of a first-in-first-out queue of a fixed length. As the current frame of event instances is added to the queue, the oldest frame is deleted. Frames store their event instances as a two-dimensional array of event instances where each array of event instances represents one level of recursion in the event type hierarchy. Event instances and potential event instances are held in the frame where they were last observed. This means that if an event instance continues to be observed, it will continue to be within the current frame. The way that an instance continues to remain in the current frame is through the event

instance replacement system of event growing and the replacement of potential event instances. Through these two systems, previous instances are deleted and replaced with a new instance that is held within the current frame of the window. It is only when an event instance does not get updated that it begins to move down the moving window towards deletion. Along with the rest of the details of the moving window, this mechanism is more precisely specified in section 4.6.3.

The window data structure is implemented in shared memory to allow access by both the recognition module and the association module. This is because both modules need to be able to track all event instances within the window and be able to note when an event instance ceases to be grown into the current frame (which means they also cease to be grown so the final length of the event instance is known).

4.3.6 Dealing with Multi-Frame Events

The three extending concepts presented in the previous three subsections allow the system to deal with the three issues of the multi-frame event instances. On-going event instances are straightforwardly dealt with through the use of event growing. Delayed event instances are handled through a combination of potential event instances and the moving window.

The final case, the case of overlapping event instances, is dealt with by a combination of event growing and potential event instances. Consider a composite event instance where the two component event instances overlap. When the first component event instance is detected a potential event instance for the composite event type is generated. For every subsequent frame where there exists the first component event instance but not the second, another potential event instance is generated, which through event growing becomes a single longer potential event instance. When the overlapping second component event is observed, the potential event instance is replaced with a normal event instance with the same start time as the potential event instance. For each subsequent frame where either both event instances or the second event instance is present, the event grower continues to grow the composite event instance. If the first component event instance stops and then starts again later with no break in the second component event instance, then a new instance of the composite event type is created and the previous instance ends. This new composite event instance is not allowed to be merged with the previous composite event instance through event growing.

4.3.7 Output Streams

As the stream of input event instances continues frame after frame, observed on-going and delayed event instances finish and so move backwards in the window. Once an event instance has left the window, there is no more possibility for it to be a component part of any further recognised event types.

This means that at that point it is no longer possible to finish recognising potential event instances that were generated on the basis of observing an event instance that was recently deleted. Therefore the predictions manifested in those potential event instances have failed. This failure should be taken into account as part of evaluating the significance of an observation. Therefore failed potential event instances where one of their constituent event instances was observed to happen are passed to the significance module to be used as evidence against the existence of that particular event type existing. These negative instances form the negative stream mentioned at the start of this section. Only potential event instances where one of the constituent event instances was observed to happen are used in this manner; event instances which were generated on the basis of another potential instance are not, as the evidence for their prediction in the first place was less sound.

All composite event instances that are observed in their entirety are positive evidence for the existence of their corresponding event type. Unlike negative evidence that can only be confirmed to be negative once their corresponding potential event instances are deleted from the window, positive evidence is confirmed the moment it is fully observed. However, every pair of event instances that are observed is evidence for the existence of that pair as an event type – including previously unobserved pairs for which no composite event type currently exists. As the pair of event instances that makes up a fully observed composite event instance has also been fully observed, it is simpler to use every pairing of observed event instances as the positive evidence as opposed to the event instances themselves being positive evidence in favour of their own existence as an event type. Therefore, the positive event instances need to be used to create every pair of event instances not just over the current frame but over the entire window. This needs to be done in order to collect evidence for the existence of delayed event instances. This pairing-up is the job of the association module. The positive event instances that are sent to the association module at each frame are those that have newly been observed in the current frame and form the positive stream which is passed to the association module for pairing.

As there are approximately n^2 possible pairs of observed event instances (for n observed event instances) there are some restrictions on the sort of positive event instance that is passed to the association module, to reduce the number of pairs. The only event instances that are passed to the association module are those where the corresponding event type has a significance measure (calculated by the significance module) above a defined threshold. This is one of the purposes of the significance measure and its threshold within the system as a whole. The primary purpose of the significance measure is examined as part of the discussion of the significance module; in short it limits the growth of new event types to only those comprising of well-founded event types and adds robustness to noise. The significance measure does not determine what is sent by the negative stream.

4.4 Module 3 – Association

The purpose of the association module is to systematically record each pairing of event instances that are temporally close enough together that, based on defined constraints, the pair of event instances can be said to be valid positive evidence in favour of the existence of the composite event type that represents that pair of event instances. This is the simplest of the modules, as its only role is to create pairs of event instances from a combination of the positive event instance stream of the recognition module that are within the specified constraints.

This section will first review those constraints, stating the reason for their existence and how the module enforces the constraints. This section will then go on to describe the details of how the pairs are generated that exploits the iterative nature of the input.

4.4.1 Constraints on association

There are four constraints that need to be satisfied for two event instances to be paired. They are as follows:

1. Both event instances must have corresponding event types that have a significance measure above the defined threshold value.

2. Both event instances must either overlap or the end of the first event instance and the start of the second event instance must both lie within a common window.

3. Both event instances must have event types that are of the same hierarchical level.

4. Both event instances must not share the same event type.

The first constraint is stated for completeness and is not enforced by the association module. It is instead enforced by the recognition module as it can be enforced more efficiently by that module. This efficiency exits because it is more efficient in this case to not transmit an event instance than to transmit it only for it to be immediately filtered out. The constraint exists to allow for the benefits that were mentioned in the discussion of the recognition module and will be explored in more detail in the discussion of the significance module – the primary benefits being that it constrains the growth of candidate composite event types to only those that comprise of event types that are strongly believed to exist and it reduces the system's sensitivity to noise, with a secondary benefit of reducing the number of instances that need to be paired.

The second constraint is based on the idea presented in the description of the recognition module (extending one of the arguments of chapter two concerning the inter-stimulus interval) that there exists a cut-off point where the delay between two event instances is so large that the effort needed to track

such a large delay is more than the value of the evidence for the existence of the compound event type is worth. This constraint is enforced by the independent operation of the moving window in shared memory – particularly the automatic deletion of event instances that fall outside the moving window. This means the association process does not need to check pairs of event instances that would not satisfy the constraint.

The third constraint makes sure that the layered nature of the hierarchical event type system remains intact. This constraint is enforced by each frame of the moving window storing its event instances in a separate array for each level. This allows for the association module to only associate the event instances that are of the same hierarchical level. This separation of levels also allows for the system to be more time-efficient.

The fourth and final constraint is checked and enforced as each pair is generated. The purpose of this constraint is to avoid the same reason the event grower is stopped from extending event instances over a delay. The constraint is in place to avoid the situation where an event that is grown over a delay may be in danger of asserting that two mutually exclusive events are happening simultaneously.

4.4.2 Event Instance Pair Generation

The recognition module passes instances of all new event instances of significant event types to the association module, including potential event instances. These new event instances are immediately paired off with every other event currently present in the window at the hierarchical level of the event under the constraints listed above. The new event instance is the second event of the pair. These pairs are stored in a list of on-going associations, a list that is divided up by hierarchical level for easier access.

At each frame the association module looks at the event instances located in previous frame in the moving window. Due to the way the window operates, all of the events in that frame are events that have just finished. Taking each finished event instance in turn, the system looks at every pair in the list of on-going associations that the finished event instance is a part of – as either component. If both event instances of that pair are not potential events, then that pair is sent to the significance module as positive evidence for the existence of the compound event made up by that pair. The pair is deleted from the list of on-going associations.

The remaining pairs involving potential event instances are allowed to remain in the list of on-going associations until both of the event instances have left the current frame of the moving window. When the pair of event instances leaves the moving window the pair is deleted, this happens whenever the section of the list they are on is next searched, or during a per-frame clean-up at the end of the frame processing, whichever happens first. As described before, when a part of a potential event instance is observed the instance moved to the current frame of the moving window, thereby only pairs that involve potential

event instances that failed to occur within the moving window end up being deleted without being passed along to the significance module.

To give a better understanding of the pairs that are generated by the association module, figure 4.5 depicts the moving window and 13 intervals representing event instances arranged to form each of the possible Allen (1983) relations with the moving window. Each interval is inclusive at both ends, so for instance, event instance intervals 1 to 8 are all three frames long. The current frame is marked as t – meaning that in the diagram event instance 8 has not yet begun and event instance 7 has only been observed once. The symbol W denotes the size of the moving window. As with the event instance intervals, the moving window is inclusive at both ends. With this input of event intervals, the association module would generate every possible pair of two event instances with the exception of three pairs. The event instance pairings that would not be generated are the pairs (1, 7), (1, 8) and (2, 8). It should be noted that the association module would generate the event instance pair (2, 7) and that pair represents the largest delay the moving window allows.

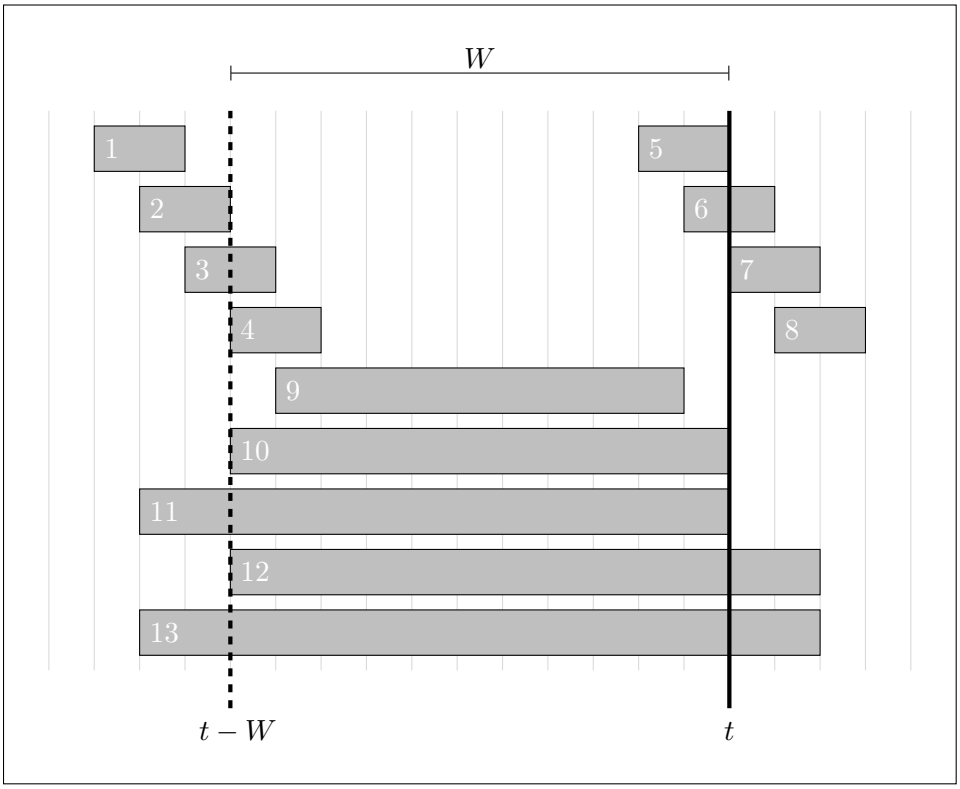

Figure 4.5: The moving window in relation to a set of event instances. Light grey lines depict a frame. Bold black lines depict the moving window – solid for the current frame and dashed for the last frame. Dark grey numbered intervals depict the event instances.

4.5 Module 4 – Significance

The purpose of the significance module is to collate all the evidence in favour and against the existence of a particular composite event type. Those composite event types where the module deems there is significant evidence for the existence of an event type are then allowed to be components of a higher-level composite event type. This module takes the most amount of inspiration from classical conditioning of all the modules, to the extent of modelling some of the phenomena of classical conditioning over several different models. These models are what determine the significance value of a particular event type.

The module is designed to allow different measures of significance to be used. These measures are referred to as models and the choice of model is a changeable setting. This means that different models of conditioning were able to be developed, each to different levels of fidelity. This allows for testing of the hypotheses of the project stated in chapter one. Both hypotheses are tested by observing the different results of each model of classical conditioning. Chapter five explores how the different results are to be compared.

This section is divided up as follows. First, the nature and effect of the significance measure is discussed. This is followed by a review of the processing carried out before and after the application of the selected model. The remainder of the section then describes each of the ten models in turn.

4.5.1 The Significance Measure

The significance measure is analogous to the concept of the association strength from classical conditioning. In classical conditioning, the role of the association strength is an indirect measure of the likelihood that a biologically relevant stimulus is about to take place. It is a measure of how much preparation an organism is investing given its subjective prediction of the event type that represents the stimulus. While the system has no ability to act on its observations, it is able to give a prediction of the occurrence of an event type, namely in the form of generating potential event instances. The significance measure associated with a potential event instance reflects how the level of belief that the potential event instance shall be observed in full.

All of the models ultimately set a single significance measure for each of the event types. The value of this measure is limited to the range[2] $0 < V \leq 1$, where V is the significance measure. The measure is updated each time new evidence is presented to the model regarding that event type. How the significance measure is calculated is down to the particular model.

There are two thresholds involved with the association strength. The upper threshold, which is set near the maximum value of one, which has previously been mentioned, decides when an event type has enough evidence that it can be used within another event instance. The lower threshold, set near the

[2]The range excludes an exact zero to avoid potential division by zero errors.

minimum value of zero, decides when an event type should be deleted from the hierarchy of event types.

While the recognition module generates potential event instances for all pairs regardless of the measure, only those with a strong significance measure are allowed to pair-up. As mentioned earlier in the chapter there are several reasons for doing this, which shall now be discussed. The reasons for the significance measure and its corresponding threshold are firstly to ensure that new pairings are based on solid evidence and secondly to add resilience to noise and limit the number of pairs of event instances that are generated at each frame. The last of these reasons is a secondary benefit and is examined properly at the end of the discussion of the recognition module.

The significance measure's most important role is to make sure only those event types that have a strong enough evidence base are allowed to contribute to the formation of new event types. The way this is achieved is through the constraint on the association module that only pairs of instances of event types that are above the significance measure are generated. The reason for this requirement is twofold. Firstly, if a composite event type has a low basis of evidence in favour of its existence, then if the two-part pattern that the event type represents is itself used in a larger pattern of event types, there would be even less basis for the existence of the larger pattern. Secondly, if it were not the case that event types needed to reach a particular standard of evidence, then this would mean that every time that a pair of event types is observed after the first observation, it would create a new event type as it would be paired with every other current event instance. Those higher-level pairs would then also be paired-up which would cause even higher-level pairs to be created. There would be no cap to the growth of the event type hierarchy and it would grow without bound.

Another role for the significance measure is in noise-reduction. The nature of the noise that is reduced is through the arguments forwarded in chapter two, namely that happenstance will naturally create random pairings of event instances. Assuming the model of conditioning employed is rational, then positive evidence of the existence of a compound event type would increase the significance measure and negative evidence would reduce it. Through the increase based on positive evidence, criterion 1 (A passive learning system needs to learn its environmental model from repeated co-occurrence of particular signal values) of the criteria presented in chapter two is satisfied. Through the reduction based on negative evidence, the significance measure satisfies criterion 7 (A passive learning system needs to undo a learned prediction in the face of new evidence that indicates that the prediction is untrue). The increase in the significance measure is analogous to part of the function of the acquisition phenomenon and the reduction in the significance measure is analogous to the function of the extinction phenomenon. These analogies further demonstrate the overall analogy between the significance measure and the association strength.

4.5.2 Non-Model Processing

To make the task of the models simpler, before and after the selected model is executed, the module processes the model's inputs and outputs in ways that are common to all models. There are two main categories of tasks. The first category of tasks is to retrieve and store the significance measure plus any other model-specific data for each input event instance pairing or negative instance. The second category of tasks is the deletion of event types that have fallen below the lower significance measure threshold.

There are two inputs to the module. The first input is a set of pairs of instances of significant event types that have been observed to have happened together and both event instances have finished (this is furthermore referred to as the positive evidence). The second input is a set of event types that were predicted to happen but the prediction has been confirmed to be a failed prediction (this is furthermore referred to as the negative evidence).

Each piece of evidence, both positive and negative, is provided to the model individually. Before each piece of evidence is provided to the model, the module retrieves the significance measure and any further measures the selected model has defined. If the measures for the positive evidence could not be found, it is because the corresponding composite event type does not exist yet. In this case of the composite event type not existing, the module creates it, setting the significance measure to be equal to the lower significance threshold. If a model requires further custom measures, it can define a function that creates and initialises those measures. It is not possible for the composite event type to not exist for negative evidence because the negative evidence was formed due to a failed expectation of the composite event type happening.

4.5.3 The Models

The purpose of a model is to calculate the significance measure, V for each event type, based on each piece of evidence presented to that model. The models described below vary in complexity and fidelity with which they imitate the phenomena of classical conditioning. The chapter introduction noted that the system presented in this chapter only focuses on a sub-set of the phenomena that were described and analysed in chapter two. While some of those phenomena can be seen in all aspects of the system, it is the models and how they control a composite event type's significance measure that draws most from the concepts of classical conditioning.

In the discussion of existing models of classical conditioning in chapter three, one categorisation applied to the models was whether a model was trial-level or real-time. In trial-level models, the computation is dealt with after an event instance has terminated. In real-time models, the computation happens at every time-frame, and can cope with those frames being arbitrarily small. The system presented in this chapter is arguably both: The system as a whole deals with real-time data, segmented into arbitrarily small frames. However,

at the level of the models used in the significance module, the system operates as a trial-level model.

There are ten models in total. The names for these models are:

1. Fixed Increment
2. Symmetrical Fixed Increment
3. Count Only
4. Absolute Acquire-Extinguish
5. Iterative Acquire-Extinguish
6. Temporal
7. Reacquiring
8. Blocking
9. Inhibition
10. Pre-Exposure

The first three models presented are in some sense, the "control" models, approaching the problem of calculating a significance measure without drawing from the concepts of classical conditioning, of which two were developed in a naïve manner. The latter seven models do draw from the concepts of classical conditioning to varying degrees. Each of the last six models of conditioning is built based on the previous model, amending it to add further phenomena.

None of the models provide any form of theoretical explanation as to how any phenomenon of classical conditioning arises within biological systems. This is deliberate and this ethos of only replicating phenomena as a black box is one of the founding hypotheses of this project, expressed in chapter one as hypothesis A. The consequence is that some of the phenomena have been developed in a highly biologically implausible manner; for example, by testing for the conditions in which the phenomenon has been observed to occur and then applying the effect of the phenomenon, rather than building a model where all phenomena are just cases of a single unified mechanism.

4.5.3.1 The Fixed Increment Model

The Fixed Increment model is the most naïve of all the models used by this thesis. As its name suggests, each time it receives a piece of positive evidence, the model returns a significance measure that has changed by a constant f^+. This is expressed in equation 4.10.

$$V_{n+1} = V_n + f^+ \tag{4.10}$$

The constant f^+ is a parameter defined prior to the system beginning to process any frames of input data and is set to be in the range $0 \leq f^+ \leq 1$. As described earlier, the significance measure is limited to the range $0 < V \leq 1$. This means that ultimately, the Fixed Increment model considers an event instance pairing to be strongly associated after a fixed number of presentations, with the number of presentations being $\frac{\tau_{\text{upper}}}{f^+}$, where τ_{upper} is the upper significance threshold.

The model does not take into account any negative evidence. When the model is provided with some negative evidence via its respective input func-

tion, it always returns a significance measure value change of zero. By having a model that does not use the negative evidence that is provided, it allows for the utility of the negative evidence to be seen.

4.5.3.2 The Symmetrical Fixed Increment Model

The Symmetrical Fixed Increment model is the enhancement of the Fixed Increment model that does take into account any negative evidence that it is provided with. As with the non-symmetrical model, this model returns a significance measure value that has changed by a constant value f^+, which is set to be in the range $0 \leq f^+ \leq 1$. When the model is provided with negative evidence, the model again returns a significance measure value that has changed by a constant value, this time by a second constant f^-, which is set to be in the range $-1 \leq f^- \leq 0$. While the model is built with two separate constants, in practice, f is set to be equal to $-f^+$. By setting the two constants to be the same magnitude with opposite signs, it allows for the Symmetrical Fixed Increment model to be more directly compared with the asymmetric version, as it gives equal weighting to both positive and negative evidence. As with the Fixed Increment model, the significance measure is limited to the range $0 \leq V \leq 1$.

4.5.3.3 The Count Only Model

The third model is based on frequentist probabilities and is the final model of the ten models in this section that is not based on classical conditioning. For each event type, including atomic event types, a count of the number of times that a particular event type has been observed is stored as supplemental data. These counts are then used to calculate a new significance measure.

When positive evidence of a composite event type is presented, the corresponding composite event type counter is incremented. If the component event types of that composite event instance are atomic event types, those event instances are counted too, but only if those particular atomic event types have not yet been observed within that frame.

When negative evidence of a composite event type is provided, only the observed atomic event types that had not yet been observed as part of any other evidence are used, and those atomic event type counts are incremented along with the other atomic event types.

Atomic event instances are counted by the model maintaining its own count for each observed atomic event type. Before the model increments an atomic event type count, the model searches for the event type in a list of atomic event types that have been incremented during that frame. Only if an atomic event type is not on the list is the count incremented. The list of seen atomic event instances is reset after all evidence produced in that frame has been processed, using the `frameTick` function discussed in section 4.6.5.

Every time a composite event type count is updated, the significance measure is recalculated for that composite event type and any composite event

types that it is a component part of. When an atomic event type is updated, only the composite event types that the event type is a part of are updated.

The significance measure for the composite event type $T_{1,2}$ composed of event types T_1 and T_2 is calculated according to equation 4.11. The derivation of the function is provided in appendix B.

$$V = \frac{2\,|T_{1,2}|}{|T_1| + |T_2|} \tag{4.11}$$

Where V is the significance value, $|T_{1,2}|$ is the count for the composite event type and $|T_1|$ and $|T_2|$ are the independent counts for the corresponding component event types. As $|T_1|$ and $|T_2|$ are independent from $|T_{1,2}|$, they also include in their count those occurrences that appeared outside any pairing with each other.

4.5.3.4 The Absolute Acquire-Extinguish Model

The Absolute Acquire-Extinguish model maintains as supplemental data for each composite event type a count of how many instances of positive evidence have been presented and how many instances of negative evidence have been presented. These two counts are both used to calculate the significance measure, which is recalculated with the presentation of each piece of evidence. There are two components to the significance measure formula, one for positive evidence and one for negative evidence.

The positive component of the significance measure formula (V^+) is known as the logistic function and is the most common sigmoid function, as shown in equation 4.12.

$$V^+ = \frac{1}{1 + e^{-k_1 \epsilon^+}} \tag{4.12}$$

Where ϵ^+ is the count of all the observed positive evidence and k_1 is a learning rate constant that determines how fast a composite event type moves along the curve. The logistic function was chosen to replicate the s-shape of the classical conditioning acquisition curve. This is not the typical function in which models of classical conditioning display the s-shaped curve of acquisition. Typically models of classical conditioning such as the Rescorla-Wagner (1972) model add a percentage of the difference between a maximum significance value and the current significance value, leading to diminishing gains in the significance measure, forming an s-line asymptote at the maximum significance value. The common method does not have a slow initial start like the logistic curve, which, as discussed in chapter two, can have uses in minimising the effect of coincidental event instance pairings. The logistic curve is however used widely within artificial neural networks.

The negative component of the significance measure formula (V^-), as expressed in equation 4.13, is a simple linear decay.

$$V^- = -k_2 \epsilon^- \tag{4.13}$$

Where ϵ^- is the count of all the observed negative evidence and k_2 is a learning rate constant that determines how fast a composite event type moves along the curve. Again, this differs from the typical equation used in models of classical conditioning in that most apply an inverted version of the acquisition function, decaying a large amount for high significance values and a small amount for small significance values. During the review of literature regarding classical conditioning, no experimental evidence could be found that suggests that extinction follows a sigmoidal decay, as is usually assumed by most models such as the Rescorla-Wagner model. The data provided by Pavlov (1927, pp. 52–53) suggests that extinction follows a roughly linear decay. This observation is reflected in the choice for the negative component of the significance measure formula.

These two component parts of the formula are combined together through addition, as shown in equation 4.14.

$$V = V^+ + V^- = \frac{1}{1 + e^{-k_1 \epsilon^+}} - k_2 \epsilon^- \tag{4.14}$$

The range of values the significance measure can take is actively enforced through checking the value and if it is out of range, replacing it with the appropriate limit of the range. The reason the range needs to be actively enforced is due to the linear subtraction of negative evidence. If there is enough negative evidence, then the significance measure would become negative, forcing it out of the range the system expects the significance measure to be in.

4.5.3.5 The Iterative Acquire-Extinguish Model

The Iterative Acquire-Extinguish model covers the same phenomena as the Absolute Acquire-Extinguish model. The difference is that the significance measure formula has been changed in order to make it iterative, in the sense that the significance measure modification can be calculated at each frame based only on its value at the previous frame and the input of the current frame. By making the formula iterative, the model does not need to maintain absolute counts of the observed positive and negative evidence, meaning the model does not need to store any supplemental data about a given event type.

Unlike the Absolute Acquire-Extinguish model, where both counts of evidence are included in the one equation, the iterative model has two separate equations, one that is applied when a piece of positive evidence is encountered and one that is applied when a piece of negative evidence is encountered. Both equations calculate the change required for the significance measure value (ΔV) given the increase in the number of pieces of evidence observed. For completeness, the relationship between the significance measure before the supply of a piece of evidence (V_n) and after its supply (V_{n+1}) is stated in equation 4.15.

$$V_{n+1} = V_n + \Delta V \tag{4.15}$$

The change in significance value given a piece of positive evidence is given by equation 4.16. In that equation k_1 is the same learning rate constant for positive evidence as the Absolute Acquire-Extinguish model; Δx is the amount one reinforcement instance is to be counted (this is normally equal to one and is only included for completeness) and all other symbols are the same as they were previously defined.

$$\Delta V = \frac{(1 - V_n)\, e^{k_1 \Delta x} + V_n - 1}{e^{k_1 \Delta x} + \frac{1}{V_n} - 1} \tag{4.16}$$

The change in significance value given a piece of negative evidence is given by equation 4.17. In that equation k_2 is the same learning rate constant for negative evidence as the Absolute Acquire-Extinguish model and all other symbols are the same as they were previously defined.

$$\Delta V = -k_2 \Delta x \tag{4.17}$$

The derivation of both significance measure change formulae are provided in appendix B.

The Iterative Acquire-Extinguish model was originally believed to produce the same results as the Absolute Acquire-Extinguish model. In testing however, the results were found to be significantly different. This was found to be due to the fact that in the Absolute model, once an association has been extinguished, it cannot be reacquired. In the iterative model, however, associations can be reacquired. This result is further discussed in chapter six.

4.5.3.6 The Temporal Model

The first expansion to the number of phenomena modelled is to include the effect of the inter-stimulus interval. This brings in the event instance timing information that is provided when positive evidence is provided to the model. This model is built upon the Iterative Acquire-Extinguish model. As there is no timing data for negative evidence, the equation for the change in significance value following negative evidence is unchanged. Due to the complexity this model adds, the discussion of how the model works is found in appendix B. What is presented here is a description of the changes that the model introduces to the Iterative Acquire-Extinguish model.

The way that the event instance timing data influences the change in significance measure for positive evidence is that through equations 4.19, 4.20 and 4.21, a single value is produced, ψ, which lies in the range $0 \leq \psi \leq 1$. This value is multiplied with the output of the previous version of the positive evidence formula. This gives the revised version of the positive evidence formula, which is shown in equation 4.18.

$$\Delta V = \psi \frac{(1 - V_n)\, e^{k_1 \Delta x} + V_n - 1}{e^{k_1 \Delta x} + \frac{1}{V_n} - 1} \tag{4.18}$$

The calculation of ψ itself is based upon two intermediate values, ϕ and χ that take the timing data of the event instances that influence the size and shape of the inter-stimulus interval curve. These intermediate values are shown in equations 4.19 and 4.20. In those equations, $t_{S,1}$ is the start time of the first event instance; $t_{E,1}$ denotes the end time of the first event instance; $t_{S,2}$ is the start time of the second event instance; $t_{E,2}$ denotes the end time of the second event instance; and W represents the size of the moving window.

$$\phi = \frac{1}{2} - \frac{1}{2}\max\left(0, \frac{t_{E,1} - t_{S,2}}{t_{E,2} - t_{S,2}}\right) + \frac{1}{2}\max\left(0, \frac{t_{S,2} - t_{E,1}}{W}\right) \tag{4.19}$$

$$\chi = \max(0, (t_{S,2} - t_{S,1} - 2\phi)) \tag{4.20}$$

The intermediate values are then fed into the function calculating the timing coefficientψ, as shown in equation 4.21. The way that all three equations 4.19, 4.20 and 4.21 are arrived at is explained in appendix B.

$$\psi = \frac{2\,(2-\phi)\,e^{\frac{-2(\ln(\chi)-1)^2}{(2+\phi)^2}}}{\chi\,(2+\phi)\,\sqrt{\frac{\pi}{2}}} \tag{4.21}$$

4.5.3.7 The Reacquiring Model

The Reacquiring model adds to the temporal model the effects of the reacquisition phenomenon. There are two effects of the reacquisition phenomenon in the way that it changes the acquisition curve each time a reacquisition phase occurs. The first effect is that the rate at which the subject regains any lost association strength is faster. The second effect is that the point at which the acquisition curve begins to level-off is higher. Both of these effects can be achieved if the asymptote of the acquisition changes proportionally to a count of the observed positive evidence.

The reason that varying the asymptote of the positive evidence curve proportionally to the positive evidence count will achieve the desired effects is because it changes the maximum height that can be achieved for a set number of positive pieces of evidence. This causes the first effect because it takes fewer positive pieces of evidence to achieve the same effect. The second effect is caused because the asymptote the function is approaching is higher.

This asymptote needs to change in proportion to the count of the observed positive evidence. The reason for this is that there needs to be a state that, when an event type has been extinguished, the state retains the fact that there have been previous times where the significance value has been at a higher point than at present. There is little point creating an iterative function based on this count, as the value stored for the iterative version would only be influenced by that absolute count, and so would purely be an indirect measure of the absolute count itself.

Because of this need for the count of positive evidence for each composite event type, the Reacquiring model and models based upon it maintain as

supplemental data a count of all the positive evidence a particular composite event type has received. The Reacquiring model does not maintain a count of the negative evidence nor does it maintain a count of any observed atomic event types.

The way that the asymptote is varied is by multiplying the absolute version of the positive evidence equation with current desired asymptote value. The absolute function with the asymptote multiplier is then treated to the same derivation of the iterative function that was used in the Iterative Acquire-Extinguish model. The resultant function is shown in equation 4.22, where the variable a is the reacquisition asymptote value and all other variables are the same as they have been previously defined.

$$\Delta V = \psi \frac{(a - V_n)\, e^{k_1 \Delta x} + V_n - a}{e^{k_1 \Delta x} + \frac{a}{V_n} - 1} \tag{4.22}$$

In order to maintain a significance value that lies between zero and one, the asymptote value itself must always lie between zero and one. This constraint can be achieved if the function to calculate the current asymptote value is itself asymptotic to one. The equation that is used is shown in equation 4.23. In equation 4.23, ε^+ denotes the count of the positive evidence; k_3 is a positive constant $k_3 \in R^+$ where the smaller the constant, the faster the value approaches one and k_4 is a positive constant $0 \leq k_4 \leq 1$ which is the initial value of asymptote, which is the minimum value that the asymptote is allowed to take.

$$a = k_4 + (1 - k_4) \left(\frac{\epsilon^+}{\epsilon^+ + k_3} \right)^2 \tag{4.23}$$

The negative evidence equation is unchanged from the Iterative Acquire-Extinguish model and the derivation of both equations 4.22 and 4.23 is described in appendix B.

4.5.3.8 The Blocking Model

The Blocking model is an extension of the Reacquiring model to add the effect of the blocking phenomenon. In this model, the blocking concept is extended from there being two predictor event types and one predicted event type to there being arbitrarily many of both. The basic inspiration for the implementation of this model is taken from the idea of subtracting from a maximum association strength, as done in the Rescorla-Wagner (1972) model.

For each piece of positive evidence, the model searches through the list of all positive evidence that has been or will be passed to the model during the current frame. The model searches to find the corresponding event type that has the largest significance value $V_{\max}$ that shares the same second component event type. The difference between the largest active significance value and the significance value of the evidence being processed is then used to define a maximum limit on the amount of change that the significance value of the

evidence being processed can undergo. This is done by subtracting that difference from the largest possible significance value allowed (i.e. a value of one). This is expressed in equation 4.24 where V_n is the current significance value of the evidence being processed and $\Delta V_{\max}$ is the largest amount of change to that significance value that is allowed.

$$\Delta V_{\max} = 1 - (V_{\max} - V_n) \tag{4.24}$$

The $\Delta V_{\max}$ value is then compared with the usual ΔV value as calculated by equation 4.22 to give a new value $\Delta V'$, which is the value that is now returned by the positive evidence function. This is expressed in equation 4.25.

$$\Delta V' = \min(\Delta V, \Delta V_{\max}) \tag{4.25}$$

It should be noted that it is sufficient to only search through the list of the entire current frame's positive evidence because all the relevant event instance pairs necessarily share the same second event instance. As that second event instance finishes second of any associated pair, all event instances associated with that event instance will be passed to the model during the same frame.

Again, the negative evidence function remains unchanged from the function in Iterative Acquire-Extinguish model.

4.5.3.9 The Inhibition Model

The Inhibition model is based on the Blocking model, extending it to add the effects of the conditioned inhibition phenomenon. Conditioned inhibition involves the creation of a different type of association. When translating the phenomenon into associations between event types, it means that for each composite event type a list needs to be maintained comprising of other event types which, if present, can contribute to an explanation for a piece of negative evidence other than the default explanation. The negative evidence exists because the association that the composite event type represents does not. This list is stored with each composite event type using the supplemental data system. With each inhibitory event type in the list, there is an associated inhibitory significance value $0 \leq U \leq 1$.

The Inhibition model modifies the negative evidence function so that every time a piece of negative evidence is received, four things happen:

1. The set of all currently potentially inhibitory event types Θ is created by looking at all the event instance pairs that the association module created that involves the component of the event type that did have an event instance.

2. From the set Θ, all the event instances that have an event type with an existing inhibitory association with the event type have their inhibitory significance values summed. The sum is subtracted from the old change in association strength formula as shown in equation 4.26 where Γ is the

set of all inhibitory associations of the composite event type, U_i is the i^{th} inhibitory significance value and all other symbols are defined as before.

$$\Delta V = -\max\left(0, k_2\Delta x - \sum_{i\in\Theta\cap\Gamma} U_i\right) \tag{4.26}$$

3. Each event instance on the list Θ that is not on the list Γ are added to that list with an initial U value of τ_{I}, which is a constant threshold which determines when an event type is to be removed from the list of inhibitors.

4. Each event type on the list Θ is given an increase in its corresponding U value, including the newly added event types. The amount each U value is increased by is done according to the iterative sigmoid curve developed in the Iterative Acquire-Extinguish model. For completeness, the function used is shown in equation 4.27. Equation 4.27 uses a different rate constant k_5 to the constant used in the standard acquisition function.

$$U_{n+1} = U_n + \frac{(1-U_n)\,e^{k_5\Delta x} + U_n - 1}{e^{k_5\Delta x} + \frac{1}{U_n} - 1} \tag{4.27}$$

The Inhibition model modifies the positive evidence function to extinguish any inhibitors that are present at the time. When the model is provided with an example of positive evidence, four things happen:

1. The set of all currently potentially inhibitory event types Θ is created by looking at all the event instance pairs that the association module created that involves the component of the event type that did have an event instance.

2. Each inhibitory event type that appears in both the set Θ and the set of all inhibitory associations of the composite event type Γ has its inhibitory significance value subjected to an extinction function shown in equation 4.28. This equation is the same as the negative evidence function for the Iterative Acquire-Extinguish model but with a separate decrease rate k_6.

$$U_{n+1} = U_n - k_6\Delta x \tag{4.28}$$

3. For each of the inhibitory event types that have a new inhibitory significance value below the threshold τ_{I} that inhibitory event type is removed from the set of all inhibitory event types Γ.

4. From the set Θ, all the remaining event types that are also present in the list of inhibitory existing event types have their inhibitory significance values summed. The sum is subtracted from the old change in

association strength formula as shown in equation 4.29 where Γ is the set of all inhibitory associations of the composite event type, U_i is the i^{th} inhibitory significance value and all other symbols are defined as before. The limit due to the blocking effects is applied after the subtraction of the inhibitor sum.

$$\Delta V = \max\left(0, \psi\frac{(a - V_n)\, e^{k_1 \Delta x} + V_n - a}{e^{k_1 \Delta x} + \frac{a}{V_n} - 1} - \sum_{i \in (\Theta \cap \Gamma)} U_i\right) \tag{4.29}$$

4.5.3.10 The Pre-Exposure model

The final model presented extends the Inhibition model to add the effects of pre-exposure of component event types prior to the first time they are associated. This adds the effects of the classical conditioning phenomena known as the U.S.-pre-exposure effect and latent inhibition. As there is little difference between the two phenomena other than the stimulus the effect works on, both effects were added simultaneously.

Both effects require knowing (or having the ability to derive) the absolute count of how many times a particular event type has been observed for all event types, including atomic and composite event types. If there was not such a record, there would be no way of knowing whether the component event types of a new event instance pair have been previously observed.

Due to the Reacquiring model needing the same value, the count of the observed positive evidence can be extended to include atomic event types using the method used in the Absolute Acquire-Extinguish model – i.e. through keeping a record of the atomic event types that had been observed during each frame so that each atomic event instance is only counted once per frame.

The way that the event type counts are used to add the effects of pre-exposure is to multiply the main part of the positive evidence function (i.e. the part from which the sum of the inhibitory significance values subtracted) by two ratios. Both ratios are the ratio between the count of the positive evidence for the composite event type and the counts of the positive evidence for one of the component event types. This change is shown in equation 4.30. In equation 4.30, $\epsilon^+_{1,2}$ is the positive evidence count for the current composite event type, ϵ^+_1 and ϵ^+_2 are the positive evidence counts for the corresponding component event types and all other symbols are the same as they were previously defined.

$$\Delta V = \max\left(0, \psi\left(\frac{(a - V_n)\, e^{k_1 \Delta x} + V_n - a}{e^{k_1 \Delta x} + \frac{a}{V_n} - 1}\right)\left(\frac{\epsilon^+_{1,2}}{\epsilon^+_1}\right)\left(\frac{\epsilon^+_{1,2}}{\epsilon^+_2}\right) - \sum_{i \in (\Theta \cap \Gamma)} U_i\right) \tag{4.30}$$

In the Pre-Exposure model, the negative evidence function is unchanged from the function used in the Inhibition model.

4.6 Formal Description and Implementation

The majority of this chapter has been dedicated to developing an intuitive understanding of how the system operates. This final section shall present the system in a more formal manner, discussing the structure of the kind data the system deals with and the algorithms that act upon that data.

The final version of the system was implemented in Java. However, an initial version was constructed in Prolog and this influenced some of the design choices. It is also why the input and output data files for the system are valid Prolog files. The reason the final version of the system was not implemented in Prolog was that Prolog did not allow for many of the performance optimisations that were required to make the system run quickly to be easily implemented. Furthermore, the back-tracking facility within Prolog proved to be more hindrance than help in this case.

The section will first look at the kinds of data the system deals with, describing the structure of that data. The four subsections following that will each look at the implementation of each of the four modules that make-up the system. The final sub-section shall describe the format of the output and how it is produced.

4.6.1 System Ontology

The system data structures can be represented as a series of tuples, which shall be described in this section. In reality, these tuples are encoded in the system's code as Java classes. The reason for representing the data as tuples instead of Java classes is to abstract away from the programming details. Note that in the pseudo-code listings in this section, the notation "$x.y$" denotes accessing the variable labelled y in tuple x.

There are eight main kinds of data the system deals with: The base objects, track-boxes for those objects, sets of track-boxed separated into frames, the derived state of a frame, event instances, event types, event association instances and the moving window. These are the primary concepts that the system deals with and are each discussed in the following subsections. Some of the types of data contain further aspects and this will be discussed as it arises.

4.6.1.1 Objects

Objects are a 2-tuple (ι, σ) where ι is an integer that represents the id of the object and σ is a string denoting the name of the object. The object id is maintained to be unique for each object name. The object id is to allow for equality checking without having to resort to string-matching. Note that in the Java code, objects are named items to avoid conflicts with Java's own internal naming system.

4.6.1.2 Track Boxes

A track-box is a 6-tuple (O, t, x, y, w, h) where O is the set of objects in that track box; t is an integer representing the time (frame number) in which the track-box exists; x and y are integers representing the two-dimensional coordinates of the centre of the track box and w and h are integers representing the width and height of the track-box. This track box data is directly created from parsing the input Prolog file, as described in section 4.2.1. It is also during this parsing that the detected object strings are assigned a unique id number.

4.6.1.3 Frames

A frame is a set of track-boxes that are present in a given frame of the input. It is a 2-tuple (t, B) where t is an integer that represents the frame's time (the frame number) and B is the set of track-boxes present in the frame.

4.6.1.4 Frame-States

A frame state represents the state of all the objects in a given frame, including the state of the spatial relationship between objects. A frame-state is a 4-tuple (t, O, S_u, S_b) where t is an integer that represents the frame's time; O is the set of all objects present in that frame; S_u is the set of all unary frame-states and S_b is the set of all binary frame-states.

A unary frame-state represents the state of an individual object at a given point in time and is a 5-tuple (t, o, ω, x, y). In the 5-tuple, t is an integer that represents the time at which the state applies; o is the object for which the tuple applies; ω is an integer that represents the area of the object and x and y are integers that represent the coordinates of the centre of the object.

A binary frame-state represents the relationship between two objects in at a given point in time and is an 8-tuple $(t, o_1, o_2, R_t, R_x, R_y, R_{dc}, R_{de})$. In the 8-tuple, t is an integer that represents the time at which the relationships hold; o_1 and o_2 are the two objects in the relationship; R_t is the topological relationship between the two objects; R_x is the horizontal orientation relationship between the two objects; R_y is the vertical orientation relationship between the two objects; R_{dc} an integer that represents the distance between the two centres of the objects and R_{de} is an integer that represents the shortest distance between the two object boundaries (the external distance). The topological relationship is one member of the enumeration $\{\texttt{DisC}, \texttt{ExtC}, \texttt{Co}\}$ where `DisC`, `ExtC` and `Co` stands for disconnected, externally connected and coalescent respectively and are described in section 4.2.2. The horizontal orientation relationship is one member of the enumeration $\{\texttt{leftAB}, \texttt{leftBA}, \texttt{inlineX}\}$ and the vertical orientation relationship is one member of the enumeration $\{\texttt{aboveAB}, \texttt{aboveBA}, \texttt{inlineY}\}$ which are both again described in section 4.2.2.

The discussion of frame-states in section 4.2.2 talks of fluents and predicates. These functions do exist, but instead of calculating the respective

properties each time that function is called, they perform a look-up in the relevant frame-state tuple. This allows for the actual calculation to be performed only once.

4.6.1.5 Event Instances

An event instance E is a 3-tuple (T, t_S, t_E) where T is the event type as described in the following subsection; t_S is an integer denoting the event instance's start time and t_E is an integer denoting the event instance's end time. There is a single sub-class of an event instance, known as a composite event instance. An event instance can either be the base type (used for atomic event types, described in the following subsection) or the composite type.

A composite event instance is a 5-tuple $(T, t_{S,1}, t_{E,1}, t_{S,2}, t_{E,2})$ where T is the event type; $t_{S,1}$ and $t_{E,1}$ are integers representing the start and end time respectively of the first component event instance and $t_{S,2}$ and $t_{E,2}$ are integers representing the start and end time respectively of the second component event instance. The start and end times of the event instance as a whole are then derived from the component event instances: t_S is the earliest of either $t_{S,1}$ or $t_{S,2}$ and t_E is the latest of either $t_{E,1}$ or $t_{E,2}$. This extra timing information needs to be recorded for use in the significance module.

The composite event instance tuple also encodes whether the event instance is a potential event instance, as described in section 4.3.4. This is done by initially setting both the start and end times to -1 for the component event type that has yet to be observed, signifying that the information is not known yet. While any of the four time variables are set to -1, then the composite event instance is considered to be a potential event instance. The pseudo-code listed in algorithm 4.1 details the function that is used to test a composite event instance to see if it is a potential event instance.

Algorithm 4.1 `isPotentialEvent`

Input:

e: The composite event instance to be tested to see if it is a potential event instance.

Output:

True if the composite event instance is a potential event instance, **False** otherwise.

```
1: if e.t_{S,1} = −1 ∨ e.t_{E,1} = −1∨
2:    e.t_{S,2} = −1 ∨ e.t_{E,2} = −1 then
3:    return True
4: else
5:    return False
```

4.6.1.6 Event Types

The event type T is a 4-tuple (l, V, D, M) where l is an integer denoting the recursive level of the event type; V is a real denoting the significance level of

the event type set by the significance module; D is a set of the composite event types that this event type forms a part of; M is a reference to any supplemental data that is stored by the chosen model of classical conditioning. Event types are polymorphic and must be one of two sub-types: Atomic event types, and composite event types. It is in these sub-types that stores the qualitative nature of the event type.

Atomic event types are a 3-tuple (ζ, o_1, o_2) where ζ is a member of an enumeration listing tokens for all the atomic event types (e.g. `approaching`, `mergeR` etc.) and o_1 and o_2 are objects to which the event type occurs. Where an event type in the enumeration only operates on a single object (e.g. `found`, `moveUp` etc.), then only o_1 is used and o_2 is ignored (though set to be the same event as o_1). Atomic event types define the recursive level l to be zero and the significance level, V, to be one. D and S are handled by the super-class.

Composite event types are a 2-tuple (T_1, T_2) where T_1 and T_2 are event types. The recursive level value l is calculated on creation of the tuple by incrementing the l value of T_1 (T_2 is enforced to have the same value). The significance value V is initialised to be equal to the lower significance threshold τ_{lower}.

The key to understanding the event-type system is that though the list D and the T_1 and T_2 values of the composite event type, the event type system forms a doubly-linked layered graph without edge weightings (Black, 2005). Each node in a layer has two parent nodes (except for the atomic event types) but can have an arbitrary number of child and sibling nodes. This is the event type database that the recognition module uses to recognise patterns of event instances and the significance module uses to maintain the state of an event pairing's signficiance data. The event type database can be accessed by two methods: Firstly, as described above, each event instance has a direct reference to its corresponding type in the event type database. Secondly, the set of all atomic event types $\mathbb{T}_0$ is maintained, this list can be used to allow the whole database to be traversed, though this is only done once, when producing the final output of the system, as described in section 4.6.6.

4.6.1.7 Event Association Instances

In the third module, event instances of different types are associated together. An event association instance records this association pairing. An event association instance is a 2-tuple (e_1, e_2) where e_1 and e_2 are event instances.

As part of the third module, there also exists a two-dimensional list A of event association instances. This list is a list of all the association instances where one or both of the component event instances has not yet terminated. It is used to compensate for the fact that different pairs of event instances will terminate at different times and to mark which pairs have been associated within the current position of the moving window.

4.6.1.8 The Moving Window

The moving window is a fixed-length queue, where each item in the queue is a set of all event instances that was observed to occur at a given time (known as a window frame, described below). The window is implemented as a 3-tuple (F, c, u) where F is a list of the observed window frames; c is an integer representing the index of the most recent window frame and u is and integer representing the number of elements of the array that are currently in use. The reason the variable u is needed is that while the queue is fixed-length, for the first few frames of input there will be unused array elements, which can affect some of the element-access calculations. Note that as also defined in other areas of this thesis, the symbol W denotes the system parameter for the maximum allowed size of the moving window.

Window frames are a 3-tuple $(E, l_{\max}, t)$ where E is a two-dimensional list of event instances that last occurred in that frame, $l_{\max}$ is an integer representing the size of the largest recursive level event instance present in that frame and t is the time that the frame referrs to. The reason that E is two-dimensional is that event instances are grouped by their recursive-level size. Both the two-dimensionality of E and the presence of the variable $l_{\max}$ are performance optimisations. As event instances are constrianed to only be associated by module three with event instances of the same level in the event type heirarchy, by grouping the event instances by their recursive level, the module can avoid reading event instances that cannot meet that constraint. Similarly, by storing $l_{\max}$, it allows for whole window frames to be skipped by the association module.

There are three operations that can be applied to the window data structure. These are required due to the shifting nature of which indices are used for that last and first frame. Algorithms 4.2, 4.3 and 4.4 each list the pseudo-code of these operations. Algorithm 4.2 corrects a given index to allow for the wrap-around in the window frame array[3]. Algorithm 4.3 moves the moving window along by adding a new current window frame and deleting the oldest. Algorithm 4.4 returns a window frame indexed by how many frames in the past it occurred. In addition, there is one window frame operation, which is required to find an event instance of a particular type. The pseudo-code of the window frame search operation is listed in Algorithm 4.5.

4.6.2 Module 1 – Pre-Processor

This first module is the most straight-forward of the modules in its implementation. As discussed before, there are two stages of processing: One stage

[3]In algorithm 4.2, it should be noted that the modulo function wraps-around with negative numbers too, something that does not happen with the "%" operator present in most languages. Because of this lack of negative wrap-around, a correction is needed, the actual operation needs to add W to i' for negative values of i only.

Algorithm 4.2 `translateWindowPosition`

Input:
i: The index position to be corrected.
W: The window size system parameter (a global-scope variable).
Output:
i': The corrected index.

1: $i' \leftarrow i \bmod W$
2: **return** i'

Algorithm 4.3 `addWindowFrame`

Input:
$\mathbb{w}$: The moving window.
$\mathbb{f}$: The window frame to be added to the moving window.
W: The window size system parameter (a global-scope variable).

1: $\mathbb{w}.c \leftarrow$ `translateWindowPosition` $(\mathbb{w}.c + 1)$
2: $\mathbb{w}.F\,[\mathbb{w}.c] \leftarrow \mathbb{f}$
3: **if** not every element of $\mathbb{w}.F$ is in use (i.e. $\mathbb{w}.u < W$) **then**
4: increment $\mathbb{w}.u$

Algorithm 4.4 `getWindowFrame`

Input:
$\mathbb{w}$: The moving window.
i: The frame to get, numbered by how many frames in the past it occurred (0=current frame, 1=previous frame).
W: The window size system parameter (a global-scope variable).
Output:
$\mathbb{f}$: The window frame requested, **null** on error.

1: $\mathbb{f} \leftarrow$ **null**
2: **if** $i < W \wedge i < \mathbb{w}.u$ **then**
3: $i' \leftarrow$ `translateWindowPosition` $(\mathbb{w}.c - i)$
4: $\mathbb{f} \leftarrow \mathbb{w}.F\,[i']$
5: **return** $\mathbb{f}$

Algorithm 4.5 `fetchEvent`

Input:
$\mathbb{f}$: A frame of the moving window.
T: The event type to find.
Output:
e: The event instance of type T present in the supplied window $\mathbb{f}$, if it was present, **null** if an event instance of type T could not be found.

1: **if** $\mathbb{f}.l \leq T.l$ **then**
2: **Repeat** (for each $e \in \mathbb{f}.E\,[T.l]$):
3: **if** $e.T = T$ **then**
4: **return** e
5: **return null**

to turn a set of track boxes into a frame-state and another stage to turn a frame-state into a set of atomic event instances. In both these states, all that happens is that each of the possible qualitative cases (state or atomic event instance) is tested for. If those cases are found to be true, the corresponding tuple is generated and added to the relevant list.

For turning a list of track boxes into a frame-state, algorithm 4.6 is used. This algorithm implies the existence of two sub-algorithms to calculate unary and binary components of the frame-state. These algorithms are not explicitly listed due to their obviousness and long-winded nature. The unary frame state simply copies the relevant information from the track box and calculates the area of the track box. The binary frame state directly applies the tests for the relations described in section 4.2.2.

Algorithm 4.6 `buildFrameState`

Input:

t: The time of the frame.

B: The set of track boxes for each object present in the frame.

Output:

s: The frame-state of the supplied track-boxes.

1: $O \leftarrow \varnothing$
2: $S_u \leftarrow \varnothing$
3: $S_b \leftarrow \varnothing$
4: **Repeat** (for every integer value of i, $0 \leq i < \texttt{size}(B)$):
5: add $B[i].O$ to O
6: build the unary state for $B[i]$ and add it to S_u
7: **Repeat** (for every integer value of j, $i < j < \texttt{size}(B)$):
8: build the binary state for $B[i]$ and $B[j]$ and add it to S_b
9: **return** (t, O, S_u, S_b)

Once the frame-state has been constructed for two consecutive frames, the two frame-states are used to calculate the atomic event instances that occurred between them. This is a case of testing the frame-states against each of the event type definitions listed in appendix A and then generating the event instances for the event types that are found to apply. The definitions in appendix A use the event calculus (Kowalski & Sergot, 1986), which is not strictly used within the system internally. The reason for this is that the use of the event calculus is considered to make the definitions easier to follow while having notational equivalence. The equivalence can be seen in the fact that the `holdsAt` predicate, which is the only event calculus predicate used in the definitions, effectively corresponds to a frame-state lookup. Once an atomic event instance has been generated, it is added to the first level of the current window frame. The recognition system then uses the atomic event instances that have been generated and added to the current window frame.

There is one part of the atomic event instance generation that needs to be explicitly stated. This is to do with the "visible" state – whether a particular object is present in a frame. This state is not computed directly but

instead needs to be inferred from the presence or otherwise of an object in the object list. When comparing frame-states to produce the atomic event instances, the two sets of objects in each frame-state are processed to build three derived sets: the intersection of the two object sets and the two possible subtraction sets. The two subtraction sets ($O_1 \setminus O_2$ and $O_2 \setminus O_1$) are used to define the lost and found event types. The intersection is then used as the set of objects to be compared against the definitions of the other atomic event types. Algorithm 4.7 creates these three derived sets.

Algorithm 4.7 `compareObjectLists`

Input:
O_1: The first set of objects.
O_2: The second set of objects.
Output:
Λ: The intersection of sets O_1 and O_2
$\Xi_{1,2}$: the set resulting from subtracting O_2 from O_1
$\Xi_{2,1}$: The set resulting from subtracting O_1 from O_2

1: $\Lambda \leftarrow \varnothing$
2: $\Xi_{1,2} \leftarrow \varnothing$
3: $\Xi_{2,1} \leftarrow O_2$
4: **Repeat** (for each $o_1 \in O_1$):
5: $\quad \eta \leftarrow$ **False**
6: $\quad$ **Repeat** (for each $o_2 \in O_2$):
7: $\quad\quad$ **if** $o_1 = o_2$ **then**
8: $\quad\quad\quad \eta \leftarrow$ **True**
9: $\quad\quad\quad$ add o_1 to Λ
10: $\quad\quad\quad$ remove o_1 from $\Xi_{2,1}$
11: $\quad$ **if** $\eta =$ **False then**
12: $\quad\quad$ add o_1 to $\Xi_{1,2}$
13: **return** $(\Lambda, \Xi_{1,2}, \Xi_{2,1})$

4.6.3 Module 2 – Recognition

The second module is the most intricate in terms of the processing that occurs. There are two primary sub-stages to the processing that occurs in the recognition module: generating composite event instances and growing event instances. In the generation stage, the composite event instances are created by iteratively matching the previous level against the known derived types. In the event instance growing stage, each of the newly generated event instances is merged with a matching version of the event instance (if one exists) that occurred in a previous frame of the window.

Algorithm 4.8 is the primary algorithm for generating the derived event types. This algorithm generates a composite event instance for each of the derived event types that have already been generated. This is done on a level-by-level basis, starting with generating instances of the derived composite event instances of the already-present atomic event instances and then generating

the derived composite event instances of those generated event instances. This continues until an iteration of the algorithm produces no new event instances.

Algorithm 4.9 is the algorithm that actually generates the composite event instance. This is listed as a separate algorithm to that listed in algorithm 4.8 to aid readability. When generating a composite event instance, there are three possible cases that need to be dealt with: One where instances for both component event instances already exist and two where one of the component event instance does not yet exist (the case where both don't yet exist would mean no composite event instance would be generated, and so does not need to be handled). The algorithm checks which case holds for the current event type and then sets the start and end times of the component event instances as required. In the cases where only one of the two component event instances has been observed, it is these instances that are considered to be the potential event instances that were discussed previously.

Algorithm 4.8 `recogniseEventTypes`

Input:
$\mathbb{w}$: The moving window.

1: $\mathbb{f}_n \leftarrow$ `getWindowFrame` $(\mathbb{w}, 0)$
2: $\mathbb{f}_{n-1} \leftarrow$ `getWindowFrame` $(\mathbb{w}, 1)$
3: $t_n \leftarrow \mathbb{f}_n.t$
4: $t_{n-1} \leftarrow \mathbb{f}_{n-1}.t$
5: $E \leftarrow \mathbb{f}_n.E\,[0]$
6: **Repeat** (while `size` $(E) > 0$):
7: $\quad E' \leftarrow \varnothing$
8: $\quad$ **Repeat** (for each $e_1 \in E$):
9: $\quad\quad$ **Repeat** (for each $T \in e_1.T.D$):
10: $\quad\quad\quad e_2 \leftarrow$ `generateCompositeEvent` $(\mathbb{f}_n, T, t_{n-1}, t_n)$
11: $\quad\quad\quad$ **if** $\mathbb{f}_n.l_{\max} < e_2.t.l$ **then**
12: $\quad\quad\quad\quad \mathbb{f}_n.l_{\max} \leftarrow e_2.t.l$
13: $\quad\quad\quad$ add e_2 to $\mathbb{f}_n.E\,[e_2.t.l]$
14: $\quad\quad\quad$ add e_2 to E'
15: $\quad E \leftarrow E'$

The second stage is to grow the event instances that were generated on a per-frame basis so that they instead occur over multiple frames. This is done by comparing the event instances that occurred in the current window frame with those in previous window frames. Where the types match, the event instances are merged together. This is done by copying the relevant times of the previous event instance on to the newly generated instance followed by deleting the old instance.

Algorithms 4.10, 4.11 and 4.12 list the event instance growing procedure. This has been split-up to aid readability. The first pseudo-code listing in algorithm 4.10 iterates through each level of each window frame, merging event instances of the same type that have occurred in the both current and a previous frame. For the first level, only the immediately previous frame has to be compared, as otherwise a contradiction may be formed in the event

Algorithm 4.9 `generateCompositeEvent`

Input:
$\mathbb{f}_n$: The most recent frame of the moving window.
T: The composite event type to generate an instance of.
t_{n-1}: The start time of the event.
t_n: The end time of the event.
Output:
e: The generated composite event instance.

1: $e_{1,2} \leftarrow$ **null**
2: $e_1 \leftarrow$ `fetchEvent` $(\mathbb{f}_n, T.T_1)$
3: $e_2 \leftarrow$ `fetchEvent` $(\mathbb{f}_n, T.T_2)$
4: **if** $e_1 \neq$ **null** $\wedge\, e_2 \neq$ **null then**
5: $\quad e_{1,2} \leftarrow (T, t_{n-1}, t_n, t_{n-1}, t_n)$
6: **else if** $e_1 \neq$ **null** $\wedge\, e_2 =$ **null then**
7: $\quad e_{1,2} \leftarrow (T, t_{n-1}, t_n, -1, -1)$
8: **else if** $e_1 =$ **null** $\wedge\, e_2 \neq$ **null then**
9: $\quad e_{1,2} \leftarrow (T, -1, -1, t_{n-1}, t_n)$
10: **return** $e_{1,2}$

instance stream – as discussed in section 4.3.3. At the other levels, the event instances are compared with the corresponding level at every frame of the moving window, as there is no possibility of forming a contradiction by growing event instances over a gap.

There is another special case for the first level, in that there are no composite event instances. Composite event instances have more data (four time variables instead of two) and so need to be merged differently. This is done using the pseudo-code listed in algorithm 4.11. The pseudo-code in algorithm 4.11 iterates through each possible pairing of instances between the two lists and if the pair has the same type, the pair is merged.

Window frame levels other than the first level are made up entirely of composite event instances. As composite event instances have more time data values, it means that they need to be merged in a different manner to the atomic event instances. Algorithm 4.12 lists the pseudo-code that does this merge over a given pair of event instance lists of the same hierarchical level. As with algorithm 4.11, the pseudo-code first iterates through every possible pairing of instances between the two lists and merges the pair if both parts of the pair are of the same type. The difference is seen in how the time values are copied. Time values should only be copied if they are not a placeholder -1 value, such a copy may overwrite a real time value. In addition, copying the end time values needs to take into account that one of the event instances may have already stopped.

It is in the activity of merging event instances together that potential event instances get confirmed to be real event instances. This is due to -1 time values being overwritten when they exist at the destination of the copy operation, but not when they exist at the source of the copy operation. By defining potential event instances this way, it means that the change between a potential

and an actual event instance is automatic, with no further processing needed beyond that which would have been needed even if potential event instances were independently defined. If an event instance reaches the end of the moving window and there still exists time values that are -1, then the event instance is passed to the fourth module as negative evidence for that composite event type existing as an event type to be associated.

After the two processing stages are complete, the module produces a list of all the potential event instances that, due to being in the oldest window frame after the growing operation, will no longer have any opportunity left to be grown into actual event instances. Algorithm 4.13 produces this list of failed potential event instances. The list is passed to the significance module to be used as negative evidence for the existence of the composite event types the failed potential event instances represent.

Algorithm 4.10 `growEvents`

Input:
$\mathbb{w}$: The moving window.

1: $\mathbb{f}_n \leftarrow$ `getWindowFrame` $(\mathbb{w}, 0)$
2: $\mathbb{f}_{n-1} \leftarrow$ `getWindowFrame` $(\mathbb{w}, 1)$
3: `growFirstLevel` $(\mathbb{f}_{n-1}.E\,[0]\,, \mathbb{f}_n.E\,[0])$
4: **Repeat** (for every integer value of i, $0 < i \leq \mathbb{f}_n.l$):
5: **Repeat** (for every integer value of j, $0 < j < \mathbb{w}.u$):
6: $\mathbb{f}_{n-j} \leftarrow$ `getWindowFrame` $(\mathbb{w}, j)$
7: `growHigherLevel` $(\mathbb{f}_{n-j}.E\,[i]\,, \mathbb{f}_n.E\,[i]\,, \mathbb{f}_{n-1}.t)$

Algorithm 4.11 `growFirstLevel`

Input:
E_{n-1}: The atomic events that were in the previous window frame.
E_n: The atomic events that are in the current window frame.

1: **Repeat** (for each $e_1 \in E_n$):
2: $\eta \leftarrow$ **False**
3: **Repeat** (for each $e_2 \in E_{n-1}$):
4: **if** $\eta =$ **False** $\wedge\, e_1 = e_2$ **then**
5: $e_1.t_S \leftarrow e_2.t_S$
6: remove e_2 from E_{n-1}
7: $\eta \leftarrow$ **True**

Algorithm 4.12 `growHigherLevel`

Input:

E_{n-1}: The atomic events that were in a previous window frame.
E_n: The atomic events that are in the current window frame.
t: The time of the previous frame.

Repeat (for each $e_1 \in E_n$):
  $\eta \leftarrow$ **False**
  Repeat (for each $e_2 \in E_{n-1}$):
    if $\eta =$ **False** $\wedge\ e_1 = e_2$ **then**
      if $e_2.t_{S,1} \neq -1$ **then**
        $e_1.t_{S,1} \leftarrow e_2.t_{S,1}$
      if $e_2.t_{S,2} \neq -1$ **then**
        $e_1.t_{S,2} \leftarrow e_2.t_{S,2}$
      if $e_1.t_{E,1} = -1 \vee (e_2.t_{E,1} < t \wedge e_2.t_{E,1} \neq -1)$ **then**
        $e_1.t_{E,1} \leftarrow e_2.t_{E,1}$
      if $e_1.t_{E,2} = -1 \vee (e_2.t_{E,2} < t \wedge e_2.t_{E,2} \neq -1)$ **then**
        $e_1.t_{E,2} \leftarrow e_2.t_{E,2}$
      remove e_2 from E_{n-1}
      $\eta \leftarrow$ **True**

Algorithm 4.13 `failedPotentialEvents`

Input:

$\mathbb{w}$: The moving window.
W: The window size system parameter (a global-scope variable).

Output:

E: The list of failed potential event instances.

$E \leftarrow \varnothing$
if $\mathbb{w}.u = W$ **then**
  $\mathbb{f} \leftarrow$ `getWindowFrame` $(\mathbb{w}, W - 1)$
  Repeat (for each $E' \in \mathbb{f}$):
    Repeat (for each $e \in E'$):
      if $e.T.l \neq 0 \wedge (e.t_{S,1} \neq -1 \oplus e.t_{S,2} \neq -1)$ **then**
        add e to E
return E

4.6.4 Module 3 – Association

As described in section 4.3.1, associations between event instances are recorded as pairs of event instances in a list grouped by the hierarchical level of the event instances involved. Once per input frame, the module conducts two processing stages. In the first stage, all the new event association instances are generated and added to the association list. In the second stage, each event association instance is checked to see if both component event instances of the event association instance have terminated. If both event instances have terminated, then the event association instance is removed from the association list and passed to the significance module as positive evidence for the event type that the two event instances form in compound.

New event association instances are built as novel event instances are encountered. What counts as novel event instances are the most recent event instances which did not have any timings modified by the recognition module. The way this criterion is detected is that novel event instances will have a start time equal to the time of the second most recent frame. Algorithm 4.15 lists the pseudo-code of the algorithm that creates a list of all the novel event instances. The algorithm assumes that Algorithm 4.10 has already executed for the current frame.

Each novel event instance is then paired with every other event instance present in the moving window that has a high enough significance measure and is at the same hierarchical level. These pairs are added to the relevant group of the list of on-going associations. Algorithm 4.14 lists the pseudo-code for the algorithm that builds the new event association instances. It is in this algorithm that the constraints listed in section 4.4.1 are applied, either implicitly or explicitly.

Algorithm 4.14 `buildNewAssociations`

Input:
$\mathbb{w}$: The moving window.
A: The level-grouped list of on-going associations.
W: The window size system parameter (a global-scope variable).
τ_{upper}: The upper significance threshold (a global-scope variable).

1: $E \leftarrow$ `newEvents` $(\mathbb{w})$
2: **Repeat** (for each $e_1 \in E$):
3: **Repeat** (for every integer value of i, $0 \leq i < W$):
4: $\mathbb{f} \leftarrow$ `getWindowFrame` $(\mathbb{w}, i)$
5: **Repeat** (for each $e_2 \in \mathbb{f}.E\,[e_1.T.l]$):
6: **if** $e_1 = e_2 \wedge e_2.T.V \geq \tau_{\text{upper}}$ **then**
7: add (e_1, e_2) to $A\,[e_1.T.l]$

The second processing stage of the association module is to detect which event association instances have reached the state where both of their constituent event instances of the association have terminated. When an event association instance has terminated, then it is added to a list of terminated instances and removed from the grouped list of on-going instances. This is

Algorithm 4.15 `newEvents`

Input:

$\mathbb{w}$: The moving window.

τ_{upper}: The upper significance threshold (a global-scope variable).

Output:

E: The list of novel events that occurred during the current frame.

```
1: E ← ∅
2: 𝕗_n ← getWindowFrame(𝕨, 0)
3: t_{n−1} ← getWindowFrame(𝕨, 1).t
4: Repeat (for each e_1 ∈ 𝕗_n.E[0]):
5:     if e_1.t_S = t_{n−1} then
6:         add e_1 to E
7: Repeat (for every integer value of i, 0 < i ≤ 𝕗_n.l_max):
8:     Repeat (for each e_2 ∈ 𝕗_n.E[i]):
9:         if ¬isPotentialEvent(e_2) ∧
10:          (e_2.t_{S,1} = t_{n−1} ∨ e_2.t_{S,2} = t_{n−1}) ∧ e_2.T.V ≥ τ_upper then
11:            add e_2 to E
12: return E
```

the task that is carried out by algorithm 4.16. The list of terminated event association instances is then passed to the significance module as instances of positive evidence for existence of the event types that the two event instances of each event association instance forms in compound.

4.6.5 Module 4 – Significance

As discussed in section 4.5, the significance module aggregates the available evidence for whether a candidate event type pairing can be considered to have a predictive relationship. The evidence is aggregated to a single value – the significance measure V. The significance module consists of two parts: A diverse set of models for how the presented evidence should be aggregated into the significance measure, and an abstraction layer that allows for different models to be selected on program launch with no programming changes needed. In addition the abstraction layer also does an amount of processing of the module's input and output that would be common to each model. The full detail of the models themselves has already been covered in section 4.5.3 and so this section will concentrate on the abstraction layer.

The abstraction layer defines five functions that each model is required to implement so that the models can be interchangeable. These five functions are listed in table 4.5.

The remainder of the subsection shall describe the processing that takes place outside any of the models but within the significance module.

The input to the system is in the form of two lists of evidence, one for the positive evidence and one for the negative evidence. The form an item on the positive evidence list takes is that of an event association instance. The form an item on the negative evidence list is that of a composite event instance.

Algorithm 4.16 `terminatedAssociations`

Input:

$\mathbb{w}$: The moving window.

A: The level-grouped list of on-going associations.

Output:

A': The list terminated event association instances.

```
1: A' ← ∅
2: t_n ← getWindowFrame (𝕨, 0) .t
3: Repeat (for each L ∈ A):
4:     Repeat (for each 𝕒, ∈ L):
5:         set t_E to be the larger of 𝕒.e_1.t_E and 𝕒.e_2.t_E
6:         if 𝕒.e_1.T.l = 0 then
7:             if t_E ≠ t_n then
8:                 add 𝕒 to A'
9:                 remove 𝕒 from L
10:        else
11:            if t_E ≠ t_n ∧ ¬isPotentialEvent (𝕒.e_1) ∧
12:             ¬isPotentialEvent (𝕒.e_2) then
13:                add 𝕒 to A'
14:                remove 𝕒 from L
15: return A'
```

Function Name	Description
`applyReinforcement`	This function is called to provide the model with a piece of positive evidence for a specific composite event type. The parameters are the composite event type being reinforced and the start and end times for both component event instances.
`applyNonReinforcement`	This is the function that is called to provide the model with negative evidence. This function has a single parameter, the composite event type that the evidence counts against.
`needsSupplementalData`	This function has no parameters and returns true if the selected model needs to attach supplemental data and false otherwise.
`defaultSupplementalData`	This zero-parameter function returns the initial state for any custom-defined significance measures that that the selected model requires.
`frameTick`	This function has no parameters and no return values. Its purpose is to be used to notify the selected model that all processing for the current frame has been completed.

Table 4.5: A list of the functions required to be defined by any significance model.

Algorithm 4.17 shows the overall processing that takes place within the module. First the positive evidence is processed, followed by the negative. As the positive evidence is in the form of event association instances, the first task is to get the corresponding composite event type for the two event instances of the event association instance. If that event type does not exist yet, then it is created. This is the task of algorithm 4.18. Note that in algorithm 4.18, the 2-tuple consisting of a 4-tuple and a 2 tuple in line 10 represents the instantiation of both the event type 4 tuple and the sub class composite event type 2 tuple.

After the corresponding event type has been either retrieved or created then the evidence is processed by the model. With the negative evidence, the event type is already available, and so when processing each item on the list of negative evidence, the composite event type is immediately passed to the model. However, after the model has processed the composite event type, the new significance level of the composite event type may be lower than lower significance threshold. If the composite event type's updated significance level is below the lower significance threshold, then the composite event type and the composite event types that are derived from it are deleted. This recursive deletion task is handled by algorithm 4.19.

Algorithm 4.17 `applyModel`

Input:

A: A list of finished event association instances (the positive evidence).
E: A list of failed potential event instances (the negative evidence).
τ_{lower}: The lower significance threshold (a global-scope variable).

1: **Repeat** (for each $\mathbb{a}, \in A$):
2: $T \leftarrow$ `getCompositeEvent` $(\mathbb{a})$
3: $e_1 \leftarrow \mathbb{a}.e_1$
4: $e_2 \leftarrow \mathbb{a}.e_2$
5: `applyReinforcement` $(T, e_1.t_S, e_1.t_E, e_2.t_S, e_2.t_E)$
6: **Repeat** (for each $e, \in E$):
7: `applyNonReinforcement` $(e.T)$
8: **if** $e.T.V < \tau_{\text{lower}}$ **then**
9: `deleteCompostiteEventType` $(e.T)$
10: frameTick ()

Algorithm 4.18 `getCompostiteEvent`

Input:

$\mathbb{a}$: The event association instance.

τ_{lower}: The lower significance threshold (a global-scope variable).

Output:

T: An event type that corresponds to the event association instance.

1: $T \leftarrow$ **null**
2: **Repeat** (for each $T' \in \mathbb{a}.e_1.T.D$):
3: **if** $T'.T_1 = \mathbb{a}.e_2.T \vee T'.T_2 = \mathbb{a}.e_2.T$ **then**
4: $T \leftarrow T'$
5: **if** $T =$ **null then**
6: $l \leftarrow \mathbb{a}.e_1.T.l + 1$
7: $M \leftarrow \varnothing$
8: **if** `needsSupplementalData` () **then**
9: $M \leftarrow$ `defaultSupplementalData` ()
10: $T \leftarrow ((l, \tau_{\text{lower}}, \varnothing, M), (\mathbb{a}.e_1.T, \mathbb{a}.e_2.T))$
11: add T to $\mathbb{a}.e_1.T.D$
12: add T to $\mathbb{a}.e_2.T.D$
13: **return** T

Algorithm 4.19 `deleteCompostiteEventType`

Input:

T: The composite event type to delete.

1: **Repeat** (for each $T_1 \in T.D$):
2: `deleteCompositeEventType` (T_1)
3: **Repeat** (for each $T_2 \in T.T_1.D$):
4: **if** $T = T_2$ **then**
5: remove T_2 from $T.T_1.D$
6: **Repeat** (for each $T_3 \in T.T_2.D$):
7: **if** $T = T_3$ **then**
8: remove T_3 from $T.T_2.D$

4.6.6 System Output

Once the system has processed every frame of input, it outputs a text file consisting of every composite event type that which has a significance value above the upper significance threshold. This subsection looks at how this is done and the format of the text file.

As the database of event types is a layered graph, this database needs to be flattened into a single list of those composite event types that have a significance value above the upper significance threshold. This is the operation performed by algorithm 4.20.

Algorithm 4.20 `getCompositeEventTypes`

Input:
$\mathbb{T}_0$: The list of atomic event types.
τ_{upper}: The upper significance threshold (a global-scope variable).

1: $\mathbb{T} \leftarrow \varnothing$
2: $\mathbb{T}_1 \leftarrow \mathbb{T}_0$
3: $\mathbb{T}_2 \leftarrow \varnothing$
4: **Repeat** (while $\mathbb{T}_1 \neq \varnothing$):
5: **Repeat** (for each $T_1 \in \mathbb{T}_1$):
6: **Repeat** (for each $T_2 \in T_1.D$):
7: **if** $T_2.V \geq \tau_{\text{upper}} \wedge T_2 \notin \mathbb{T}_1$ **then**
8: add T_2 to $\mathbb{T}_2$
9: add T_2 to $\mathbb{T}$
10: $\mathbb{T}_1 \leftarrow \mathbb{T}_2$
11: $\mathbb{T}_2 \leftarrow \varnothing$
12: **return** $\mathbb{T}$

Once the database of composite event types has been flattened, it needs to be encoded into a format that is both human and machine readable. For this purpose, the event types are encoded into a Prolog format. This format expresses the composite events in terms of an atom predicate and a function. The atom predicate is `happensTogether` (T_1, T_2, V) which expresses the belief that event types T_1 and T_2 are believed to have a predictive relationship with significance value V.

The other component of the format, The `compositeEvent` (T_1, T_2) function, expresses the pairing of the two event types T_1 and T_2. Formally, the function takes the input event types and outputs a singular composite event type, but this definition is not required to understand the output format of the system. As with event types in general, the function is used in a recursive manner to build the patterns the composite event type represents.

These two functions are combined together to describe the format that the system outputs each individual composite event type in. An example of the how this is done is shown in equation 4.31 for a composite event would be at level two in the event type hierarchy. In the example, T_1 to T_4 represent atomic event types and V represents the significance value. Note that the stacking of parts of the line is purely stylistic, to allow the example to fit

within the page. In order to differentiate this output format with the internal system's representation of a composite event, an individual line of this output is referred to as a "rule".

$$\texttt{happensTogether}\left(\texttt{compositeEvent}\left(\begin{array}{l}\texttt{compositeEvent}\,(T_1, T_2)\,,\\ \texttt{compositeEvent}\,(T_3, T_4)\end{array}\right), V\right) \tag{4.31}$$

4.7 Chapter Conclusion

This chapter has described the system that was designed and built to test the hypotheses that were stated in the first chapter. The system comprises of four modules that pass data between each other, recognising event types and creating pairs of their corresponding event instances. When a pair of event types is observed enough times, that pair becomes a new event type to be recognised and paired up which allows for events of further complexity to be learned.

The system makes regular use of the ideas of classical conditioning within the learning process. This ranges from the concept of pairing of events for association, to the explicit implementation of the phenomena within the significance module. It is through the use of these ideas that the methods the system uses for dealing with noise, and for learning in a computationally efficient manner, were developed.

This system is evaluated in the next two chapters by exposing it to three separate learning scenarios. Each of the learning scenarios is based around the atomic event types recognised by the pre-processor module. Chapter five describes how the system was evaluated and chapter six then provides the results of that evaluation.

Chapter 5

Evaluation Methods

Chapter four presented a system that is built to test the hypotheses stated in chapter one. This chapter looks at the specific testing that was done using that system. By testing and characterising the system's ability to learn patterns of event types given different significance models, the hypotheses that the system was based upon are tested.

In order to test and characterise the patterns of event types the system can learn, the tests presented by this chapter are in the form of three learning scenarios. Each learning scenario involves a set of objects moving, based on a kinematic system that is observable within the real world. The movements were created with a program built to simulate the kinematics of each system. Bounding boxes were provided to each object based upon the object's position, current size and distance from the scene observation point.

In order to characterise how each model learned, different lengths of data were generated. Each generated dataset of each length for each scenario was then inputted into the system frame-by-frame for each of the ten models presented in section 4.5.3. The composite event types that were above the upper significance threshold at the end of each processing run (i.e. those composite event types that the system has a high level of belief in their existence) were then used as the output data from the system. The output datasets were then compared with a manually created proxy ground truth, one for each learning scenario, which listed the composite event types that were expected to be learned based on the knowledge of that learning scenario. The statistics employed in the comparison make up the first set of results that test and characterise the patterns of event types that the system can learn.

The other result set is based upon the observation from the analysis made in chapter two, that compensating for the noise of happenstance is a primary driver for many of the phenomena of classical conditioning. In order to test this idea, further input datasets were created. For each of the scenarios at

each length of video, several datasets were created that added various levels of noise to each of the input bounding boxes. The noise that was created simulated the typical sort of noise that would be present in a visual object tracking program.

The typical sort of noise present in a visual object tracking system does not at first appear to be related to the noise of happenstance discussed thus far. However, due to the way the atomic events of the system presented in chapter four have been defined, the typical kind of noise found in a visual object tracking system becomes the noise of happenstance. This is due to the noise distorting the position and shape of each object changing the relationships between objects, and between objects and the scenario. This would in turn create new atomic event instances that were not a part of those event types that define a scenario and would also mask event instances that were. For example, the slight vibration of the boundaries of a track box due to tracker noise can cause two stationary objects to appear to be rapidly and frequently approaching and receding from each other.

By comparing the output data between a noisy version of a scenario and the corresponding non-noisy version of the scenario, a characterisation of how each of the significance models and the system as whole is able to handle noise can be created. This characterisation would test the idea that some of the phenomena of classical conditioning exist to reduce noise.

For a while, it was considered that the input data to the system should be from real videos being tracked. The reason this method was favoured was because it was feared that by simulating the data, it would create the risk of introducing a confirmation bias. After spending considerable time on building a suitable tracking program it became apparent that it was infeasible using this method to create the quantities and varieties of data needed to give a reasonable characterisation of the system. In addition, the approach of using a genuine tracking system would not allow for characterisation of the system's noise handling capacity. As for the issue of confirmation bias, the stance was taken that while vigilance is always needed, the risk of confirmation bias could never be eliminated – for instance there is a risk of a confirmation bias in the scenarios that were selected. It was also deemed that in this case the risk of there being a confirmation bias in the simulator is reduced by using scenarios that are based on very widely-known and understood kinematics.

This chapter is divided as follows. First the learning scenarios that will be used are described, along with the scenarios that were considered, but for various reasons were rejected. This will be followed with a description of the simulator program that produces the data for each of the three learning scenarios. The next section will look at how the proxy ground truth for each scenario was created. Following on from that there is a section detailing how the output of the system was analysed. Next there is a description detailing how the various system and significance model parameters were set and the values used in the evaluation.

5.1 The Learning Scenarios

Due to the existing state of the system, there are two constraints that needed to be taken into account when selecting a learning scenario. The first constraint is that the target knowledge to be learned within the scenario must involve the external motion of the objects within the scene. This constraint exists due to the selection of atomic event types within the system solely describing external motion.

The second constraint is that the system in its present state would only be able to learn deterministic patterns – those patterns of event types that almost always happen due to a causal relationship existing that either one event type causes the second or that a third (potentially hidden) event type causes both component event types of the composite event type. This is because of the system not implementing some of the phenomena of classical conditioning. As briefly touched upon in chapter two, there also exists two further types of pattern; patterns that only exist if some (potentially hidden) state holds and patterns that are stochastic in their nature. If all the phenomena that were argued to contribute to learning these types of patterns (those phenomena that contribute towards criteria 8 a–d and criterion 9: Reacquisition, Spontaneous Recovery, Partial Reinforcement, The Partial Reinforcement Extinction Effect, Conditioned Inhibition and Configural Cues) were implemented then this constraint should not apply, as argued in the analysis in section 2.4.

When selecting the scenarios, six candidate scenarios were considered and are listed below. Scenarios one to three are those that were selected and scenarios four to six were not.

Scenarios that were selected

1. A person throwing a ball up into the air. The concept here is that the system would learn to expect that when the ball went up into air that the ball at some point would return.

2. Two objects rotating around a common axis. This was the test scenario that was presented by dos Santos *et al.* (2009), whose work formed the basis for the first module of the system. In this scenario the system would be learning a larger pattern of the objects approaching, crossing each other and then receding from each other. This scenario would allow the project to be related to the event type pattern recogniser presented in dos Santos *et al.* (2009) by being effectively the pattern learning system dual of their recognition system.

3. Pairs of balls and multiple balls colliding, such as those found in the game of pool. This was thought to teach the system the pattern of a ball approaching another ball and making contact would cause both balls to be moving after the contact was made.

Scenarios that were not selected

5. A fragile object falling and breaking into multiple pieces. The act of a single object splitting into multiple other objects would make use of the `lost` and `found` atomic event types. The system would learn that by losing the single object would mean that multiple other objects would then be observed. This was not selected because it would need some form of symbol-level generalisation to deal with the variable number of pieces.

6. Two people passing a ball between each other. In this scenario, it was thought that the system would be able to learn the general pattern of the ball being passed. This would in theory allow it to detect similar pass event type in other ball games such as basketball. This scenario would also allow the system of this thesis to be related to the work reported in Bennett *et al.* (2008), which the system of this thesis used for its input format. The work reported by Bennett *et al.* (2008) was evaluated using a basketball scenario, though did not do any pattern learning on that video. This was not selected because the system would again need some form of symbol-level generalisation (i.e. where the system allows for object typing rather than just unique object identifiers) in order to recognise the event type occurring given any two people, rather than the specific two people of a given video.

7. Rigid objects bouncing around an enclosed space in the style common to many screensaver programs. The concept that could be learned is that of the configuration space[1] of an object, which could be learned for arbitrary shaped-objects in an arbitrary enclosed space despite only having access to the rectangular bounding box. It would in principle be possible to represent the configuration space as a set of event type patterns – for example the system could be used to form predictions of an imminent change of direction near a container side (this would have needed a minor extension to the set of atomic event types recognised). This scenario was not selected because of this needed representation of the configuration space would make its interpretation and evaluation unfeasibly complex.

5.2 Scenario Simulation

The simulator program produces the simulated tracker data that forms the input data for the system. This section describes how that input data is

[1] A configuration space is a concept from the field of commonsense knowledge as part of knowledge representation. A configuration space is the space that describes where an object is allowed to be located and orientated based upon the existing location and shape of the object and the shape of its surroundings (Davis, 1990, pp. 282–286; Lozano-Peréz, 1983).

generated. The simulator takes three input parameters: The scenario desired, the amount of noise that should be applied to each object as a percentage (the reason it is a percentage is explained later) and the length of output needed in seconds (the program converts this to frames assuming 30 frames per second).

The data sets that were generated were for each combination of the three learning scenarios, six different lengths of video (1, 2, 5, 10, 15 and 30 minutes) and seven different noise percentages (0%, 5%, 10%, 15%, 20%, 25% and 30%).

In simulating the tracker data, there are two stages: Motion simulation and tracker simulation. The motion simulator as its name suggests applies an animating function to change the positions of the various objects present within the scenario's scene. The tracker simulator then takes the object position data and transforms this data into the sort that can be expected a tracking system would produce, including the noise due to tracker error.

Before describing each of these stages, in order to improve the realism of the simulation, many stages of the simulator, both during motion simulation and tracker simulation require the use of a Gaussian random number generator that has been truncated at both tails. In order to save repeating the same information, this number generator is described here and then the main description of the simulator shall refer to the generator without description. The Gaussian random number generator is a random number generator where the cumulative distribution curve matches that of a Gaussian cumulative distribution curve. The random number generator has four parameters, the mean, the standard deviation, the minimum value and the maximum value. The minimum and maximum values define the truncation level. The Gaussian curve is ensured through a Java library function that provides a random number which when called several times cumulatively creates a Gaussian curve with a mean of zero and a variance of one. This number is then multiplied by the desired standard deviation squared and then added to the mean. If the resultant random number lies outside the range of acceptable values, then the process is repeated until a number is selected that lies within the acceptable range. The reason for the truncation is to allow for strict definition of the range of acceptable values.

5.2.1 Motion Simulation

Motion simulation is based within a scene. A scene is composed of a set of two-dimensional objects and a set of animation functions that are applied to a subset of the scene's objects. An object is initially made up of five parameters: one for position and one for size for each of the two spatial dimensions and a layer depth parameter that is used to calculate occlusion. Each object is assumed to be a rectangle permanently orientated with the axes. Each scene is able add further parameters to an object such as mass.

An animation function calculates the change in motion and size for each object it is defined to apply to. For each frame, each motion function is passed its set of objects. Outside of the motion functions, the motion is not interfered

with in any way. The implication of this non-interference is that any collision detection or handling of objects that have moved outside the visible volume of the scene need to be dealt with by the motion functions.

The remainder of this subsection will look in turn at each of the specifics of simulating each of the scenarios.

5.2.1.1 The Throwing Scenario

The throwing scenario is animated using equations and constraints derived from the standard Newtonian equations of motion to create a simulation of the motion of a person throwing a ball in the air. There are two objects within the scenario – the ball and the thrower. The objects have two extra parameters each: a component of velocity for the horizontal and vertical dimensions. The complexity of this scenario is that, to add realism, each throw of the ball is thrown at a random angle with a random start point and a random apex, with each random element being generated using a truncated Gaussian distribution.

In order to improve intuition regarding what is being simulated before explaining how it is simulated, a selection of frames from the output of the throwing simulator, figure 5.1 and figure 5.2 are a selection of frames captured from the output video of the throwing scenario simulator. The noise added by the tracker simulator discussed in section 5.2.2 has been disabled so that the tracker noise can be discussed separately. Figure 5.1 shows a sample frame from the simulation. In the frame, the ball (represented by the blue box) is in mid-air having just been thrown upwards by the person (represented by the red box). The green outlines represent the track boxes for each object.

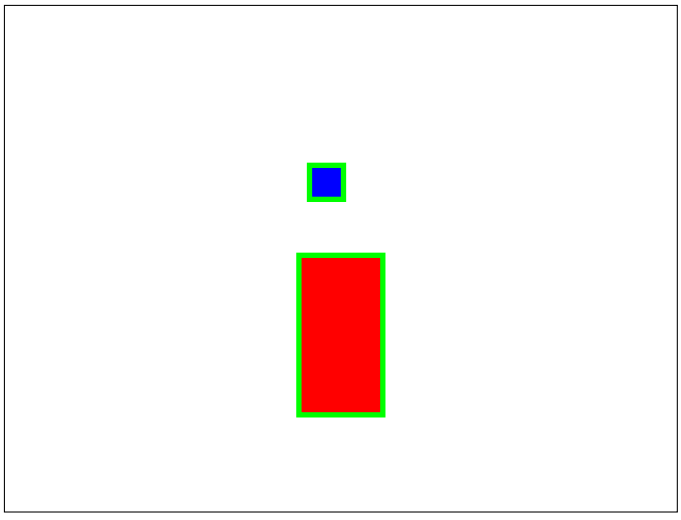

Figure 5.1: A sample frame from a video depicting the throwing scenario.The box representing the person is coloured red and the box representing the ball is coloured blue. The green outlines represent the track boxes for their respective objects.

Figure 5.2 shows a representative selection of 12 frames that occurred within a set of approximately 30 frames that happened within one second of the output video. This figure exists to show the kind of movement that occurs

in the output video. There are several features of the simulation that are worth highlighting. The first is that the ball can randomly have a small horizontal motion, as can be seen in the figure by the ball moving right. Secondly, the size of the box changes in size horizontally, tracking the ball's horizontal movement, to simulate the person stretching their arms out to catch the ball. This is then reversed in the last few frames as the person returns the ball to their core ready for another throw. The final feature worth highlighting is that because the simulator uses the Newtonian equations of motion, there are far fewer frames in the middle of the ball's flight than those near the apex of the flight. This final feature may not be wholly apparent from the selection of frames. This is because, as mentioned earlier, over half the frames in the sequence were left out for the sake of making the differences in consecutive frames more noticeable.

Figure 5.2: A selection of frames depicting the kind of movement found in the throwing scenario. The box colours are the same as in figure 5.1.

The remainder of this subsection describes how this motion was simulated. There are four input parameters to the throwing scenario: the acceleration due to gravity, the throwing acceleration due to the thrower, the catching deceleration due to the thrower and the maximum apex height of throw. The parameters for the throwing acceleration and catching deceleration may be modified if the chosen parameters are unable to allow the motion of the ball to fit within the constraints for the apex of a throw.

There are three phases for a single throw: the initial acceleration of the throw, the ball under the influence of gravity alone and the deceleration of the catch. Once the ball has reached a velocity of zero, the first phase begins again.

The height of the thrower defines a number of other important heights. The height of the thrower defines the highest release height, the lowest release height, the highest catch height, the lowest catch height and the lowest possible height that the ball can be decelerated to zero. The thrower's height defines

the highest release height because the release point cannot be higher than the height of the thrower's reach, which is set at 1.25 times the thrower's starting height. The thrower's height defines the lowest apex height (and so the lowest release height) because the ball's track box should always cease to overlap the thrower's track box, this is defined to be 1.25 times the thrower's starting height plus the height of the ball. The thrower's height defines the highest catch height again because this must be within the thrower's reach. The thrower's height defines the lowest possible height that the ball can be decelerated to zero because the ball must not go below a height that would not be reachable by the throwers arms – defined to be 0.25 times the thrower's starting height. The lowest possible height for the ball to come to a stop due to deceleration then defines the lowest possible catch point due to the fixed deceleration.

It is these heights, the maximum apex height, the deceleration/acceleration due to gravity and the integral nature of time that defines the constraints on the throwing acceleration. The throwing acceleration must at least be large enough so that the maximum apex height is able to be reached by the longest possible acceleration time – which is constrained by the distance between the lowest height the ball can be at zero velocity and the highest possible release height. The throwing acceleration must be also small enough that the smallest possible amount of acceleration (1 frame's worth) does not put it over the highest apex point. If the throwing acceleration is outside these constraints, then it is set to be the nearest value that satisfies the constraints.

The constraints on the catching deceleration are similarly derived. The minimum catching deceleration is defined to be the amount of deceleration needed to bring the ball to a halt at the minimum halt height from the maximum catch height given the ball reached the maximum apex height and was accelerated by gravity. The maximum catching deceleration is defined to allow the constrained throwing acceleration enough distance between the stop point and the highest release point for one frame's worth of acceleration. Again, if the catching deceleration is outside the constraints, then it is set to be the nearest value that satisfies the constraints.

The random nature of the apex of each individual throw is controlled by varying the release height of the ball and so controlling how long the first phase lasts defines the velocity of the ball at release which defines the apex. The release height is chosen at random from a truncated Gaussian distribution with a minimum and maximum release height as previously defined, a mean that is half way between the two truncation points and a standard deviation that is half the difference between the two truncation points. The random nature of the stop point of each individual catch is controlled by varying the catch height of the ball in the same manner to the way the apex of a throw is controlled but with the corresponding minimum and maximum catch points.

When watching a person throwing a ball, each throw is never directly upwards. There is always at least a small amount of movement in the horizontal

axis. This is simulated in a similar manner to the vertical, though is simpler to calculate as there is no acceleration or deceleration of the ball due to gravity. The width of the thrower determines the extreme left and right distance the ball is allowed to travel between release and catch, which are defined to be the centre of point of the thrower plus or minus two times the width of the thrower. Using a truncated Gaussian distribution, a point within these two extremes is selected using a mean equal to the centre point of the thrower and a standard deviation that is equal to the difference between the two extreme points. This selected point is then used as the point that the ball is caught on the horizontal. After the ball is caught it is decelerated so that it returns to the centre of the thrower. This means that unlike the vertical calculations, the horizontal throw acceleration and catch deceleration are not provided as parameters but are instead calculated. The throw acceleration is calculated to create the correct velocity so that it is at the correct point horizontally when it reaches the chosen vertical catch height. The catch deceleration is calculated so that the ball decelerates to zero horizontal velocity when it reaches the centre of the thrower and that it reaches that point at the same time as the vertical velocity reaches zero.

The final part of the simulation is changing the size and shape of the rectangle representing the thrower to simulate the thrower reaching for the ball. This is done by having an original size and position and a current size and position. For the vertical reach simulation, if the release height is above the original height of the thrower, then the height of the thrower is increased so that the top edge of the thrower and the top edge of the ball remain the same until after the release point. While the ball is only acting under gravity, the height of the thrower is gradually increased up to the maximum height. When the ball is moving downwards and when the top edge of the ball goes past the top edge of the thrower, regardless of when the ball is caught, the thrower's height is reduced to keep the two top edges the same until it has been reduced back to the original height.

The thrower's width is also changed to simulate the thrower reaching for the ball and tracking it as it moves horizontally. The width of the thrower is continually changed such that when the ball is flight, it is always directly above some part of the thrower. However, when the ball is caught, the thrower's width moves back towards its initial width, simulating the thrower returning the ball to the core of their body.

5.2.1.2 The Rotating Scenario

The rotating scenario is the simplest of the three scenarios presented. The rotating scenario shows two objects of the same size rotating around a fixed rotation point that is equidistant from both objects. Both objects have the same constant angular velocity and are always opposite each other. The objects rotate on the x-z plane, meaning that from the point of view of the observer the objects move towards the viewing pane and then away from the

viewing pane. The layer depth parameter described at the beginning of this motion simulation section is re-interpreted in this scenario is as a depth coordinate. The scenario adds several extra parameters to each object: the distance from the focus point, the angle of how far around the rotation the object is and the angular velocity (in angles of rotation per frame).

As with the throwing scenario simulator, to improve intuition regarding the kind of motion being simulated, this subsection will first discuss a selection of frames from the output of the simulator. Figure 5.3 shows a single frame of the output video from the rotating scenario simulator. In the rotating scenario, the two objects are rotating clockwise (as viewed from above) around a central point. The frame in figure 5.3 was captured as the red object is receding from the viewpoint and the blue object is approaching it.

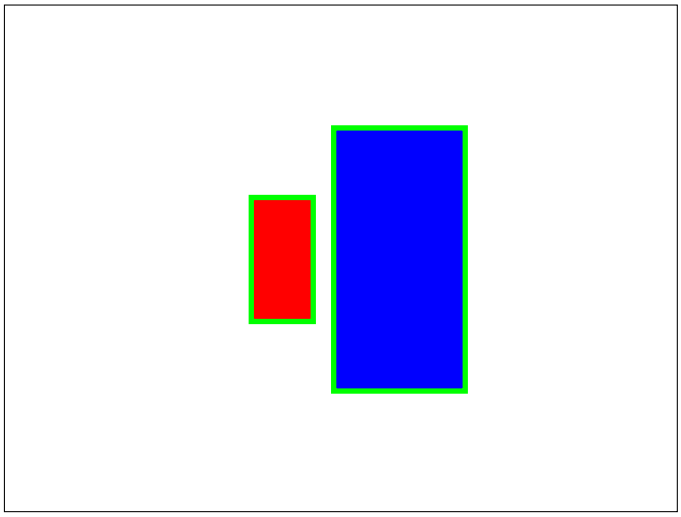

Figure 5.3: A sample frame from a video depicting the rotating scenario. The first object is coloured red and the second object is coloured blue. The green outlines represent the track boxes for their respective objects.

Figure 5.4 depicts the kind of motion that occurs within the rotating scenario. The frames are again a representative selection of 12 frames from 30 that occurred over a one second period of an output video of rotating scenario simulator. There is no variation in the motion of the objects between video outputs. The simplicity of the motion is due to the provenance of the scenario. As discussed in section 5.1, the rotating scenario was chosen as it was a test scenario used by dos Santos *et al.* (2009). Due to this origin, the motion of the objects in this scenario was designed to closely match the corresponding scenario considered by dos Santos *et al.* (2009), which entailed restricting the variability of the motion in the scenario.

The motion function has three parameters: the position coordinates of the rotation point that the objects rotate around. At each frame, the angular position is increased by its angular velocity. If the angular position is increased above 2π radians then 2π is subtracted from the angular position. When the angular position of an object is updated, the absolute position coordinates are also updated in line with the change of angular position.

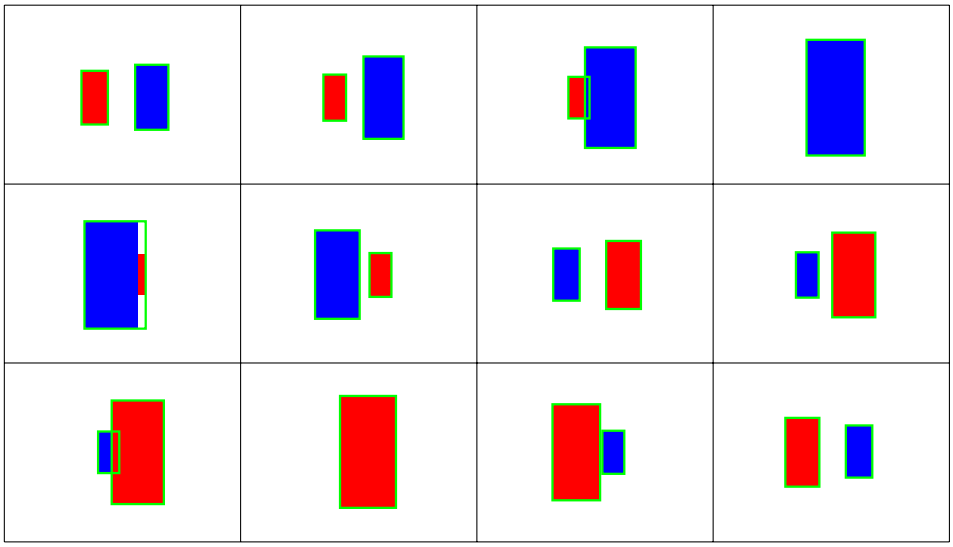

Figure 5.4: A selection of frames depicting the kind of movement found in the rotating scenario. The box colours are the same as in figure 5.3.

The value of the depth coordinate influences both the observed width and height parameters, in order to allow nearer objects to appear larger than those that are further away. This is done by defining a larger depth value to mean that an object is further away from the viewing pane and dividing the width and height parameters of each object by its depth parameter to give its corresponding coordinates and size for the two dimensional scene expected by the tracker simulator.

5.2.1.3 The Collision Scenario

The collision scenario involves four balls on a (non-tracked) table with a border. At the start a randomly selected ball (not selected using a Gaussian distribution) is given an initial impulse force with a random (non-Gaussian) angle and random (Gaussian) force magnitude. The ball is then allowed to move under friction, colliding with other balls and the border of the table. When all balls have come to a standstill, another ball is randomly selected to be given another impulse force. This continues for the whole time of the scene.

The intuition for this scenario is that of a game such as pool or snooker. Figure 5.5 depicts a frame captured from the output video of the collision scenario simulator. In the frame, the red ball was recently struck in the direction of the orange ball and has just glanced the pink ball out of the way. There are two points regarding the collision simulation that need to be highlighted. The first point is that the table does not have a green outline as it is not a tracked object. The second is that due to thickening the green outlines so that they are visible in a print format, it may not be noticeable that the balls objects are circular, rather than the rectangles used in the other scenarios. The reason for using a circular object representation is to produce an accurate simulation of the collision dynamics. Because of the tracker simulator still tracking the

objects as rectangles, there can be instances as shown in the figure between the red and pink balls where there is some overlap of the track boxes. This case is one of the reasons that the box overlap merging process described in section 5.2.2 has an overlap tolerance.

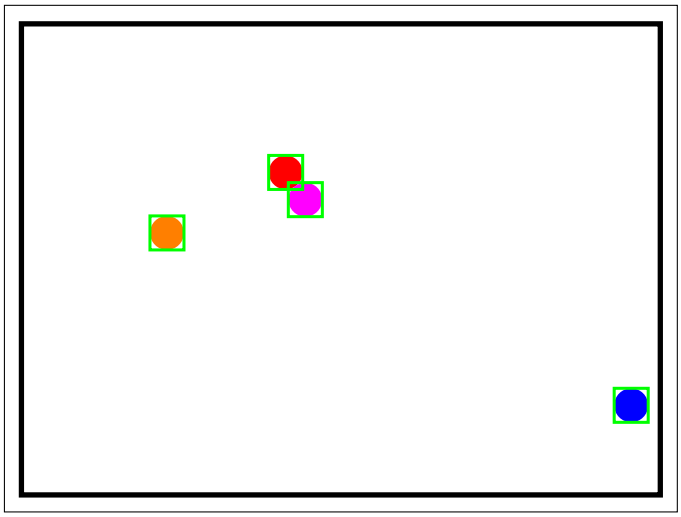

Figure 5.5: A sample frame from a video depicting the colliding scenario. The balls are coloured red, blue, pink and orange. The black rectangle represents the border of the table. The green outlines represent the track boxes for their respective objects.

Figure 5.6 depicts the kind of motion produced by the collision scenario simulator. As with the other two scenarios, the figure depicts 12 representative frames from a sequence of 30 frames that occurred over 1 second. The depicted motion demonstrates the fidelity of the motion simulation. The glancing blow between the red and pink balls shows that after the blow, the pink ball is slowly moving in a direction nearly perpendicular to the original motion of the red ball, as would be expected by a glancing blow between two balls in a game of pool. In addition, by the time the red ball has reached the orange ball, it doesn't have the momentum to move the orange ball. This part of the figure's example output showing a loss in the red ball's momentum demonstrates two further aspects of the collision scenario simulator. Firstly it indicates the simulation of kinetic energy losses due to friction and secondly it demonstrates the simulation of imperfect collision elasticity, in this case from the glancing blow between the red and pink balls.

Each ball has four extra object parameters: the radius of the ball, the mass of the ball and an x and y velocity component. In this scenario the balls are not treated as rectangles but as spheres, though the size of the bounding rectangle is maintained in line with the ball radius. The motion function has six parameters: The elasticity of collisions between balls, the elasticity of collisions between a ball and the edge of the table, the initial impulse force mean and standard deviation, the percentage of velocity lost in each frame due to friction (the friction inefficiency) and the number of sub-frames to compute.

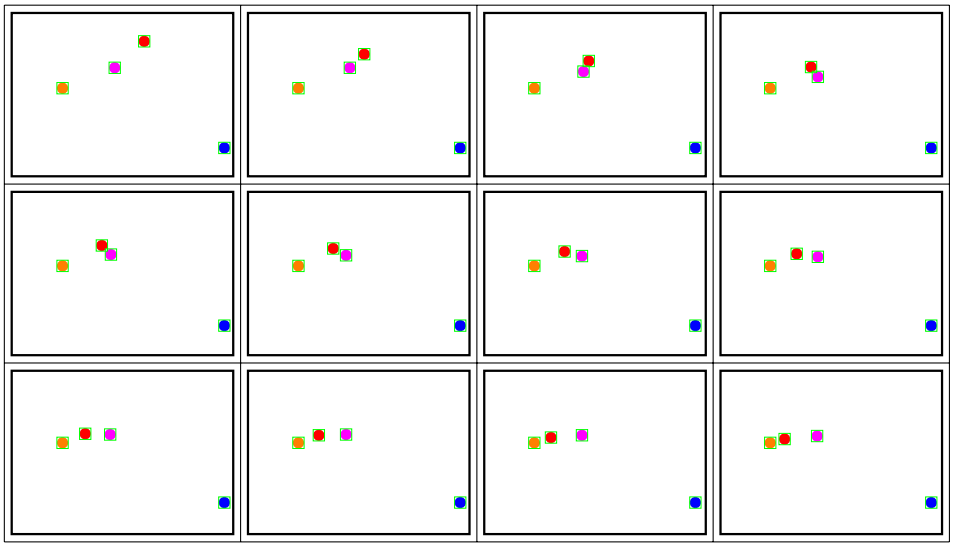

Figure 5.6: A selection of frames depicting the kind of movement found in the colliding scenario. The box colours are the same as in figure 5.5.

The collision detection employed is relatively simple. At each sub-frame, each ball is moved along by its velocity vector. Each ball is then compared with each other pair-wise. When the distance between two objects is smaller than the sum of their radiuses, a collision has occurred. This is the reason for a sub-frame parameter. In order to make sure that a ball does not move so fast in a single frame that it moves through a collision event instance and at the end of a frame is no longer overlapping the other ball.

When a collision event instance is detected, two stages of computation happen. First, both balls of the collision event instance are moved back along their current velocity vector so that the balls are in the position where the actual collision event instance took place. This is done proportionally to the magnitude of the velocities. The second stage is that the velocity vectors of each ball are changed based on the momentum of each ball, the angles of incidence, and collision elasticity.

The collision detection and resolution between a ball and each edge of the table is similar but simpler to that of two balls colliding. A collision is detected when the distance between the centre of the ball and the edge is less than the radius of the ball. When there is a collision, first the ball is moved back along the velocity vector to the position that the collision took place as before. Second, the component of the velocity vector that caused the ball to cross an edge is multiplied by -1, which has the effect of reversing the direction of the vector, and then the total magnitude of the vector is multiplied by the table edge collision elasticity parameter.

The final calculation that is computed in each sub-frame, after ball movements and collisions are detected and resolved, is the efficiency due to friction. Each ball's velocity magnitude is multiplied by the friction inefficiency parameter, ensuring that all the balls that are moving eventually come to a rest. The reason that the effect of friction is expressed as a per-frame movement in-

efficiency percentage rather than with coefficients of friction is to reduce both the number of parameters and the amount of computation required.

5.2.2 Tracker Simulation

The tracker simulator uses the data produced by the motion simulator and manipulates the data to simulate the sort of noise that a video object tracker adds during the tracking process. The tracker simulator is independent of any of the three scenarios. There are three stages in the tracker simulator: the addition of noise, the grouping of overlapping objects and output of the final data.

For each object, only the two-dimensional rectangles parameters are taken into account. This means that for each object in the current frame, there are four pieces of data to be held: the observed two-dimensional coordinates and the observed two-dimensional size values. For each object, before the noise is added, each of those four pieces of data is copied so that the data can be modified without changing the original simulated values. This copied data is referred to as a track box.

In order to define the noise that is added to the track boxes, there needs to be some qualitative understanding of the typical sources of noise that are found in visual object trackers. Tracking errors are the result of uncertainties in where and how large the bounding track box of an object should be. The observed causes of this uncertainty are: video compression artefacts, similarity of foreground and background, occlusion and the tracker using a poor model of the tracked object.

Video compression artefacts and foreground-background similarity both cause position uncertainty in the same way, by making the boundary of the object hard to detect. When the boundary of the object is hard to detect, then it may create small errors where the edge of a track box is placed. This misplacement in that edge will cause the size and position to be incorrect. As video compression artefacts and the background change, then so can the detected boundaries.

Occlusion causes some or all of an object to be not directly visible in the observed scene. In the case of partial occlusion, there is uncertainty regarding the true size of the object, and so what the size and position of the track box should be, especially if the object is moving or can change shape. This uncertainty introduces noise, as the estimate for how far into the occlusion the track box should go[2] is likely to be incorrect. In the case of full occlusion, then neither the position nor the size of an object can be ascertained with any certainty – the tracker must exclusively use expected trajectories for both size and shape. Again, as there is an estimate involved in where an object may be while under full occlusion, that estimate can be erroneous.

[2]Some trackers only try to track the non-occluded parts of an object. In the context of the discussion, these trackers always produce incorrect tracks.

The final observed source for error is the employment of a poor model of the tracked object. Consider searching for a dark blue object on a light blue background – if the model of the tracker has quantised the colour space too coarsely, then it may not be able to make the distinction between the two shades of blue. More generally, the model of the tracked object informs the tracker as to what features of a scene to look for. If those features are either not well-defined enough or too prescriptively defined, then the tracker will be looking for the wrong features and probably incorrectly identify a part of the scene as being a part of the tracked object, when in fact it is not (which is referred to as a phantom detection). Depending on how poor the model is, the errors that can be caused range from minor positional and size errors to not being able to detect the object at all.

Now that the possible errors that can be found in a tracker have been enumerated, the way that these errors are simulated can be described. The tracker simulator assumes that model errors are restricted to minor size and positional errors, that occlusion between two tracked objects is dealt with by merging the track boxes and there is no occlusion between any tracked objects and any non-tracked objects (as introducing this would entail changes to nature of the selected scenes). This means that all observed errors due to the tracker are perturbations in the track box position and size which are caused by uncertainty in each track box boundary edge and this perturbation rarely strays too far from the true value. Figure 5.7 shows the effect of adding tracker noise in all three scenarios.

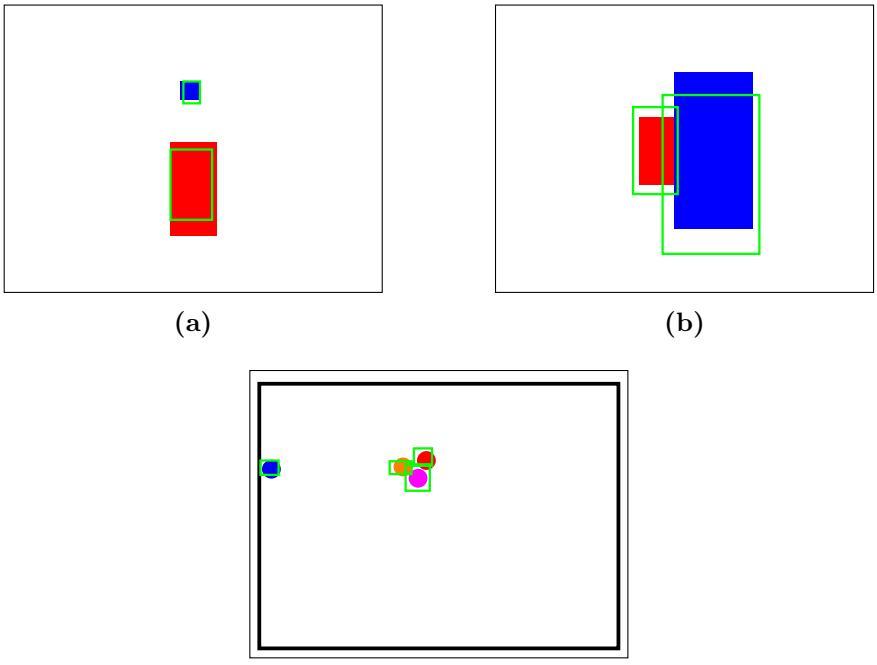

Figure 5.7: The effect of adding track-box noise to each of the three learning scenarios. The noise level in all three of these examples is 20%.

The perturbations in the track box position and size can be modelled by moving each edge of a track box independently by a random amount each

frame, with the random quantity being selected using the truncated Gaussian random number generated described earlier. This Gaussian distribution is how the noise percentage simulator parameter is used. The noise percentage simulator parameter defines the minimum, maximum and standard deviation of the distribution. For the left and right edges, the noise percentage is of the object's width. For top and bottom edges, the noise percentage is of the object's height. The absolute portions of the object's width and height are added and subtracted from each edge as appropriate providing the minimum and maximum allowed perturbation and so the minimum and maximum points on the Gaussian distribution. The standard deviation for each distribution is defined to be the appropriate absolute portion of either the width or the height. The random perturbation is selected and applied to each track box edge to give noisy track box data.

By setting the noise as being relative to the object size rather than an absolute noise level, it simulates the fact that in many kernel tracking systems, the perturbations are greater for larger objects. This is because during a pixel-level foreground-background segmentation, there can be many small clusters of pixels that are incorrectly classed as being foreground. If these clusters are near to an object, then they can be incorrectly classed as being part of that object. This has the effect of perturbing the size and position of the object's calculated boundary. With larger objects, there is a greater risk of this effect occurring as there is a greater chance of an erroneous foreground pixel cluster occurring near the object's boundary.

The next step in the tracker simulator is the grouping of overlapping track boxes. Each object's track box is compared pair-wise with every other track box. Where there is a significant overlap between the two track boxes, the two track boxes are merged into a single track box. The reason there needs to be a significant overlap is to mimic the tolerance observed in tracking software, such as the tracking software reported on by Bennett *et al.* (2008). The way this tolerance is included is by measuring the area of the overlap and comparing that to the total area of each track box. If the area of the overlap is greater than or equal to half the size of the smallest object then the two track boxes are merged. The result of the merging of two track boxes contains both object labels and the borders of the single track box are set to encompass the borders of both original track box. An example of the result of this merging process, which occurs regardless of noise level, can be seen in figure 5.4. The third frame of figure 5.4 shows an object overlap that is not large enough so that the object track boxes are merged and the fifth frame shows an object overlap that is large enough so that the object track boxes are merged.

Finally, the remaining track boxes are written to an external file as a frame of data. The output data is in the input format used by the system, which is described in detail in chapter four. The process of simulating another frame of motion then beings again and the loop continues until there are enough frames of track box data recorded to the output file to fill the specified time.

5.3 Proxy Ground Truth

To gain a measure of the quality of the results produced by the system, there needs to be data that the results can be compared to. In the absence of another system that produces comparable output, the data needs to be gained through a manual process that uses the output of human intellect as the data by which the quality of the output of the system is judged. This output is known as the ground truth. However in this case, the complexity of creating such a ground truth would make the task completely infeasible, and so only an approximation can be created. This thesis calls this approximation the proxy ground truth. This section discusses how such a proxy ground truth is created.

Using a manually produced proxy ground truth has pitfalls, as it is effectively a form of introspection, as discussed in chapter one. The very nature of data produced by humans is subjective, even if the subject matter at hand is emotively neutral. Another pitfall, related to the first, is that a human produced proxy ground truth may not be entirely complete: There may be other output data, that when the human is shown that data they would agree to knowing, but would not think to produce that datum without prompting. Finally, there may also be constraints on what data a human can consciously express due to the tractability of the task at hand. Therefore there needs to be a more objective method of creating a proxy ground truth that involves the employment of human intellect in a more systematic and structured manner than simply giving a human the task of producing the proxy ground truth.

A systematic approach has been used by this thesis to produce a proxy ground truth. The proxy ground truth takes the form of a list of composite events that would be expected to be in the output of the system. This was generated by using a five stage process: two human stages and three deterministic stages. For each scenario, a set of expected key-frames for that scenario are produced by a human. A key-frame is a frame that a human deems to be qualitatively distinct from the previous key-frame. In the next stage, these key-frames are then used to create the frame state predicates used by the system. In the third stage these key-frame states were turned into atomic event instances that occur between the key-frames. In the fourth stage, a human gives each atomic event instance an expected duration which are then placed onto a Gantt chart. The final stage then deterministically compiles each pair of atomic event instances that are within a fixed timing to give a set of composite event instances, which are placed on a new Gantt chart. The final stage is repeated using the Gantt chart of the composite events to produce higher-level Gantt charts of composite event instances. The repetition continues until either there are no new pairs of composite events or the number of levels of the Gantt charts matches the system's maximum level of event instances.

The proxy ground truth for each scenario will now be looked at in turn. The first level Gantt charts for each of the scenarios is presented in appendix C. The higher-level charts are not included within this thesis because their size can run to hundreds of rows. In the throwing scenario, seven key-frames were

found, as shown in figure 5.8. This created 42 frame states, six per key-frame. In turn this created 12 different atomic event types, each with one corresponding instance. These were compared with the same window size of six frames as used by the system (the way this parameter was set will be described later in this chapter in section 5.5), producing 64 pairs of atomic event instances. These pairs in turn created 2,248 pairs of second-level pairs for a total count of event type pairs of 2,312. No further levels were created for this or any other scenario for reasons that will be discussed in chapter six.

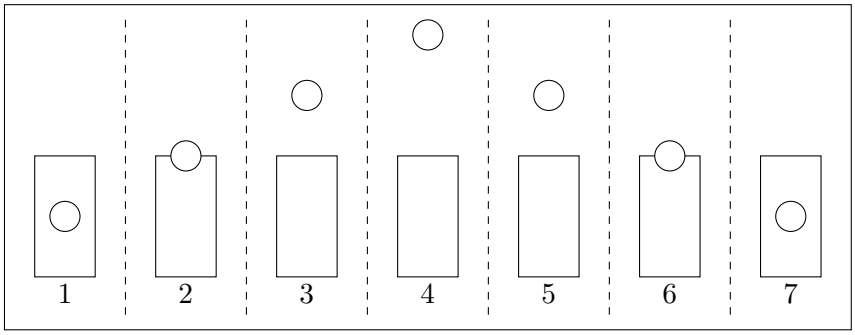

Figure 5.8: The key-frames of the throwing scenario. The circle represents the ball and the rectangle the person. The first frame is the person accelerating the ball. The second frame represents the person releasing the ball. The third frame represents the ball moving away from the person. Frame four shows the ball being stationary its apex. The fifth frame represents the ball approaching the person. The sixth frame shows the person catching the ball. The seventh frame is the person decelerating the ball.

In the rotating scenario, nine key-frames were found as shown in figure 5.9. This created 54 frame states. In turn this created 24 atomic event instances of 22 different atomic event types. These instances compared with the same window size of six frames, producing 130 pairs of atomic event instances. These pairs in turn created 6,013 pairs of second-level pairs for a total count of event type pairs of 6,143.

The colliding scenario needed a slightly different approach to the creation of key-frames. This is because twelve independent sequences of event types can occur: eight sequences that involve a pair of balls and four sequences that involve one ball. The first four two-ball sequences depict the first ball colliding with the second from each of the four primary directions. The second four two-ball sequences depict the first ball colliding with the second for each of the four diagonal directions. The four one-ball sequences depict the ball colliding with the edge of the table – though note that the table itself is not a tracked object. Each of these twelve sequences has three key-frames. The sequences at this point do not take into account that there are four balls in the scenario; the sequences just use a single ball or single pair of balls. Each of the sequences and their key-frames are shown in figure 5.10 for the sequences involving pairs of balls and figure 5.11 for the sequences involving a single ball.

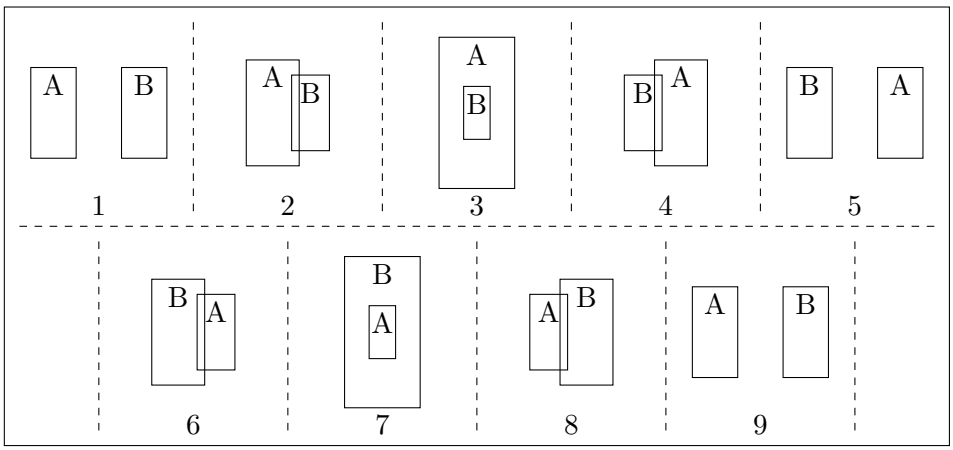

Figure 5.9: The key-frames of the rotating scenario. The objects A and B are rotating counter-clockwise around a central axis that runs parallel to the vertical dimension. This has the effect of each object in turn approaching the observer and then receding from it.

Analysing the key-frame sequences created 120 frame-states over all of the sequences. In turn this created twelve Gantt charts (one for each sequence) with a total of 112 atomic event instances of 26 different atomic event types. These instances were compared with a window size of six frames, producing 253 unique pairs of atomic event instances. These pairs in turn created 9,647 pairs of second-level pairs giving a total count of event type pair rules of 9,901 (recall from section 4.6.6 a rule is a compound event written in the system's output format). As said before, the total event type pairs currently generated are for a single ball or pair of balls. Therefore, the rules that were created need to be populated with every combination of pair of balls. When this is done, the final count of the total number of rule pairs becomes 106,218; this is because there are twelve combinations of balls for rules that involve pairs of balls and four combinations for rules that involve a single ball.

5.4 Performance Analysis

The end goal of the evaluation, is to characterise how well the system performed. In this case there are four approaches to distinguishing the performance of the system:

- How does each output of the system compare with the proxy ground truth?
- What is the qualitative nature of the output of the system?
- How well does the system cope with noise?
- How fast does the system run?

The first two approaches are considered to be the primary approaches. This is because they attempt to characterise the worth of the final output,

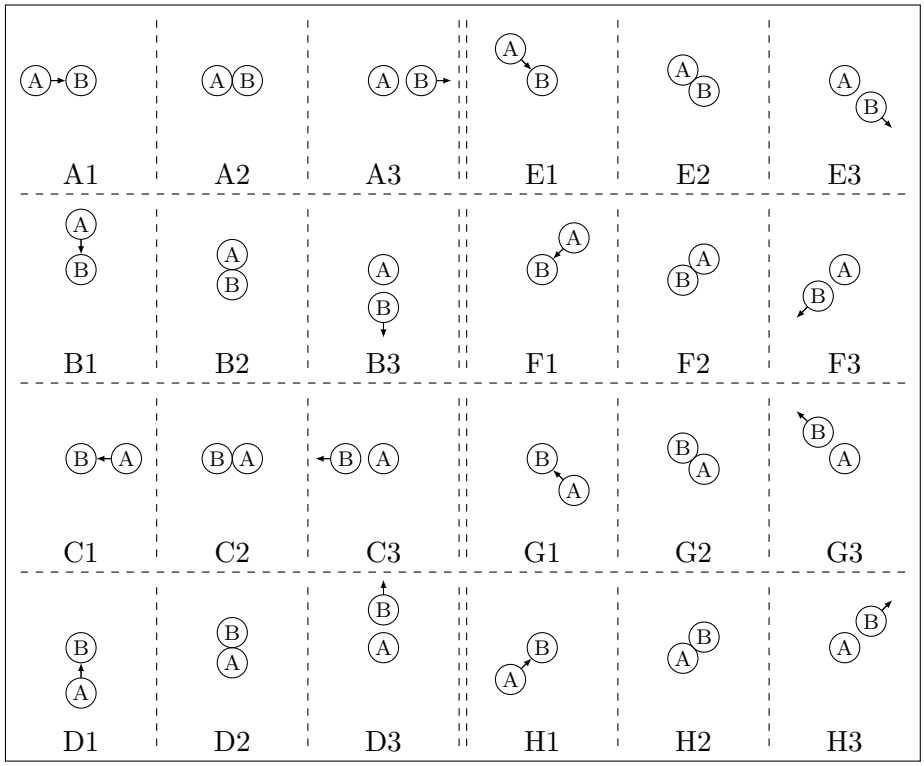

Figure 5.10: The key-frames for the eight two-ball sequences of the colliding scenario. Arrows depict the direction of movement where relevant for understanding the key-frame.

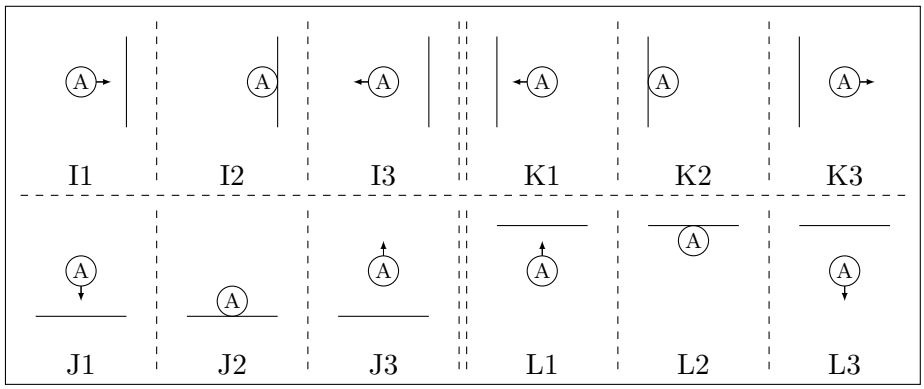

Figure 5.11: The key-frames for the four one-ball sequences of the colliding scenario. Arrows depict the direction of movement where relevant for understanding the key-frame.

whereas the other two approaches are more concerned with how that output is arrived at. Each of these four approaches requires some level of analysis so that the interpretation of the results produces well-founded conclusions. The first, third and fourth approaches are quantitative in nature, whereas the second approach is qualitative in nature. Each of the three quantitative approaches was applied to each triple of the three input parameters of model, video duration and video noise. This section describes the methodology of the three quantitative approaches. The methodology used for the second approach is explained alongside the results themselves in section 6.3. This is because the justification for the methodology choices of the second approach made relies on some of the other results.

The first and third approaches involve comparing two sets of data. This is obvious in the case of comparing the results against the proxy ground truth, but the same is true for characterising how well the system and model responds to noise. The way that the system responds to noise is measured by comparing the output of the system with the noise present against the output of the system when the noise is not present but had the same selection of model and video duration input parameters. Comparing the system's output for a noisy input against the system's output for a 0% noise input is preferred over comparing the system's output for a noisy input against the proxy ground truth. This is because it would be very rare for the proxy ground truth to be in full agreement with the output of the results, whereas if the system is very insensitive to noise, then it could be in full agreement with the zero-noise output, as this was also produced by the same system. This allows the issue of sensitivity to noise to be evaluated independently from any bias in either the proxy ground truth or any systematic bias in the output of the system.

The motivation for measuring how well the system and the models cope with noise is based on the analysis of classical conditioning presented in chapter two. In that chapter, it was argued that many of the phenomena of classical conditioning examined exist to reduce the influence of noise on the associations that the agent learns. By characterising the nose robustness of the system and the various significance models implemented within as part of it, this idea that classical conditioning phenomena exist to counteract noise can be tested. If the idea is correct, then it would be expected that the significance models that implement a greater number of the phenomena of classical conditioning would have better noise tolerance.

Answering the question as to how fast the system runs requires that the output of the system not only provides the list of rules that are above the upper significance threshold but also provides several other metrics. In addition to the list of significant rules, the system recorded the time it took in seconds to process the provided video using the selected model. This measure is not the measure that was analysed however. A larger level of input data will naturally take longer to process, so instead of reviewing the total time taken, the number of frames in the input video was divided by the total time taken to give an

estimate of the mean frames per second. This is the metric that is discussed when presenting the results.

The remainder of this section discusses the methods used for comparing of two sets of rules, as used in the first and third approaches. These comparison methods appear to be well established within machine learning literature. The methods are included within this thesis for completeness and so that it is known which evaluation methods of the range used within the machine learning literature were employed. The methods used are used for the evaluation of a classifier or other information retrieval system.

When comparing two sets of results, one set is used as the standard and the other is then compared against it. In this comparison there are four types of rule that may be present, two types of agreement and two types of disagreement. The two forms of agreement are called a true positive and a true negative. A true positive result is a rule that appears in both sets. A true negative result is a rule that appears in neither set. The two forms of disagreement are termed false positive and false negative. A false positive result is a rule that does not appear in the standard set of rules but does appear in the comparison set of rules, this concept is related to the type I error used in hypothesis testing. A false negative result is a rule that does appear in the standard set of rules but does not appear in the comparison set of rules and is related to the type II error used within hypothesis testing. All four types of result can be presented in a diagrammatic form known as a confusion matrix as shown in table 5.1.

		Standard set rule inclusion	
		Included	**Not included**
Comparison set rule inclusion	**Included**	True Positive (TP)	False Positive (FP)
	Not Included	False Negative (FN)	True Negative (TP)

Table 5.1: The confusion matrix for comparing two sets of rules.

From the size of each of type of result, the measures that are used can be derived. The first of these are known as precision and recall. Precision is the fraction of true positives against the size of the comparison set. Recall is the fraction of true positives against the size of the standard set. These can be calculated through the sizes of each of the types of result and are shown in equation 5.1 and equation 5.2.

$$\text{Precision} = \frac{|TP|}{|TP| + |FP|} \tag{5.1}$$

$$\text{Recall} = \frac{|TP|}{|TP| + |FN|} \tag{5.2}$$

From these measures, a further single measure that combines and balances the two individual measures of precision and recall can be created, called the

F or F_1 measure. The F measure was first described by van Rijsbergen (1979) and is shown in equation 5.3.

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \cdot |TP|}{2 \cdot |TP| + |FP| + |FN|} \tag{5.3}$$

The final measure used to compare two sets of rules is known as the Matthews correlation coefficient (MCC). The MCC is a correlation measure between predicted and observed classifications of a binary classifier and is shown in equation 5.4. The MCC was first described by Matthews (1975) based on a correlation measure described by Fisher (1958).

$$MCC = \frac{|TP| \cdot |TN| - |FP| \cdot |FN|}{\sqrt{(|TP| + |FP|)\,(|TP| + |FN|)\,(|TN| + |FP|)\,(|TN| + |FN|)}} \tag{5.4}$$

The MCC requires a measure of the true negative count. This value is not able to be recorded by any of the result sets because the set it represents is by definition not listed. Therefore the true negative count needs to be calculated from the values that are known. This can be done by first calculating the total size of all possible results (N) and then subtracting the known counts as shown in equation 5.5.

$$|TN| = N - |TP| - |FP| - |FN| \tag{5.5}$$

The size of all the possible results for each learning scenario is defined as the list of every possible pair of event types for each level. This can be calculated based on the number of tracked objects in the scenario and the number of atomic event types that the system detects. The throwing and rotating scenarios both use two objects and the colliding scenario uses four objects. As the system detects 8 unary, 2 reversible and 16 non-reversible atomic event types, this gives 50 possible atomic event types for two objects and 236 possible atomic event types for four objects. A possible pairing is any selection of two separate atomic event types. This means that there are $n^2 - n$ possible pairs of event types at a particular level in the event type hierarchy if there are n event types in the level below. This gives a count of 2,450 level one event pairs for two objects and 55,460 level one event pairs for four objects. This in turn produces 6,000,050 level two event pairs for two objects and 3,075,756,140 level two event pairs for four objects. These level one and level two counts summed together produce a total size of all possible results of 6,002,500 for two objects and 3,075,811,600 for four objects. The count of the number of atomic event types is not included in the totals as these are not produced as results of the system as they are not rules to be recognised.

5.5 System Constants

Over the whole system including each of the various models there are 11 constants that determine the behaviour of the system above the four input parameters of:

- The scenario.
- The input tracker noise percentage.
- The input video duration.
- The significance model to use.

Most of the constants are specific to each model, but there are some system-wide constants. This section looks at how the values for each constant were arrived at.

Each constant affects the utility of other constants – for instance, the reinforcement learning rate, k_1, affects the number of rules that can be subject to the extinction process controlled by non-reinforcement learning rate, k_2. This means that the only way to guarantee that an optimal combination of settings has been chosen is to try every combination. This is not feasible though, as to try 8 possible values for each of the 11 constants in the context of each model gives 2.73×10^8 combinations. Even running the maximum observed frame rate of the system of approximately 3000 frames per second[3] and each combination was tried using only a one minute duration video with no noise as input, it would take over five years of constant processing to process every combination. Therefore, in the interests of feasibility, each setting had to be dealt with as if it independently affected the results. Further, it was not feasible to perform a full analysis of the results as described in the previous section; the only measures used were those directly output by the system: The processing time and the number of rules in the output. While each model has its own separate constants, due to later models being based upon earlier models, many of the model parameters have equivalents in other models. In order to improve the feasibility of determining a value to each constant, each constant was reviewed only once, in the model that first uses that constant.

For each constant, the two measures were collated and plotted over approximately eight possible settings. The final setting was set based on what appeared to be the best balance between the number of rules produced and the time taken to produce them. Appendix D lists the values that were used for each constant.

---

[3]This is an extreme frame rate. A more typical frame rate would be approximately 100 frames per second; the lowest extreme is approximately one frame per second.

5.6 Chapter Conclusion

This chapter has described how the system presented in chapter four was evaluated. The system was evaluated within the context of three different learning scenarios: A ball being thrown into the air, two objects rotating around a common axis and four balls colliding with one another. These three scenarios were generated as simulations based upon the physics equations relevant to each scenario plus a method of adding noise to the simulation. The output of the simulations was input into the system. The output of the system was compared in two ways. Firstly the output was compared against a proxy ground truth that was created for each scenario based upon a process comprising of human and deterministic decisions. Secondly where the input of the system included a level of noise, the output was compared against the output that was produced for the same video without the noise. The comparisons were based on several widely-used methods. However, the same widely-used methods were not feasible for determining the large number of constants within the system and so a weaker form of analysis was used to determine what value should be used for each constant.

The results of the comparisons produced in this system are presented in the next chapter, chapter six. In the final chapter, chapter seven, those results are then used in the context of the hypotheses presented in chapter one to form a set of conclusions about the ideas presented in this thesis.

Chapter 6

Results

This chapter presents the results of the evaluation that was discussed in chapter five. Along with the results, the chapter analyses the salient points of the results. This analysis also allows the discussion to present an interpretation of which phenomena used in the system and its accompanying models worked well and those that worked less well.

During the evaluation of the system, a flaw in its design was discovered. A description of the flaw and the work-around introduced to reduce its impact is the topic of the first section of this chapter. This is presented first because it influences the remainder of the results. The second and third sections look at the primary results – the system's ability to produce a model that matches the description of what the system was supposed to learn in each scenario. This is done quantitatively in section two through a comparison between the output of the system and a proxy ground truth. The third section reviews the output of the system qualitatively.

The fourth and fifth sections look at the secondary results. The fourth section looks at how the system and its models respond to different levels of noise in the input data. The fifth section considers the computational performance of each model. The chapter then ends with a wider discussion of the results. Note that so as to not break-up the text into very small chunks, spoiling flow, in each section that presents results in the form of charts, those charts are presented at the end of that section, after the accompanying discussion text.

6.1 The System Design Flaw

While the system was processing input data, it was found that due to the interaction between the event type hierarchy and an unforeseen aspect of the extension for multi-frame events, there was a combinatorial explosion in the number of event types at each level. The combinatorial explosion meant that

the system was unable to finish processing all but the shortest of video durations. This was rectified by capping the number of levels of the event type hierarchy at two levels of composite event types, creating a hierarchy of three levels when atomic event types are included.

The reason for the combinatorial explosion is due to a combination of the way the event-type hierarchy is represented and the situation of three or more multi-frame events that always occur together in a manner that means that they always overlap one another and all the overlaps fit within the size of the window. This can cause three more event types to be created at the next level up, which will lead to three event types at the level above that and will continue forever. This concept is shown in figur 6.1, which depicts a type configuration of event instances that for the purposes of this discussion occurs many times causing event types as shown. In the case of the example in figure 6.1, a third level would consist of the composite event type tuples $((T_1, T_2), (T_1, T_3))$, $((T_1, T_2), (T_2, T_3))$ and $((T_1, T_3), (T_2, T_3))$. When the results were being processed, not just three-way overlaps were observed, but even higher order overlaps – for example, in the case of the colliding scenario, some twenty-way overlaps were observed.

In theory, the infinite chain of levels is amortised by the fact that each new level has to wait for the events of the level below to reach the significance threshold. However, this issue is enhanced by a further factor. Three-way overlapping event types cause a combinatorial explosion when other event types are associated in serial with them. When a further event type is associated serially with the three overlapping event types, six event types are created at the next level. Each event type that gets associated with the overlapping events effectively becomes a multiplier for the number of event types created, increasing the overall count for each higher level. This is demonstrated in figure 6.2. When this multiplier effect is combined with the infinite hierarchy and the larger numbers of atomic events used in the learning scenarios, it does not take very many frames or levels before the speed of processing one frame becomes too slow to feasibly process every scenario at each video duration and noise level. This led to the enforcement of a limit to the type hierarchy of two levels of compound events.

When developing the system, it was assumed that there were a finite number of levels that could be created in the event type hierarchy. The reasoning was based on the fact that one-on-one serial associations and associations between two-way overlaps would both only produce a single event type the next level up. This would mean that every level would have fewer event types than the level below it. Eventually each pairing and pairing of pairs and so on would combine to a single high-level pairing. The reduction in the number of event types at each level would thusly cause the number of levels to be finite.

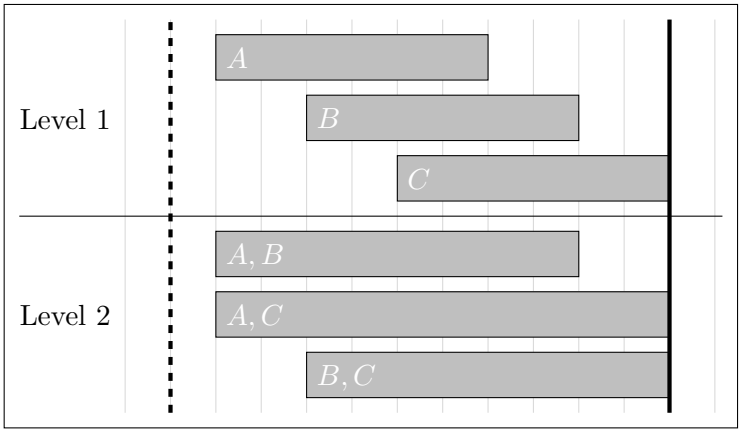

Figure 6.1: Two levels of multi-frame events. The window is marked by the two black vertical lines – solid for the current frame, dashed for the last frame.

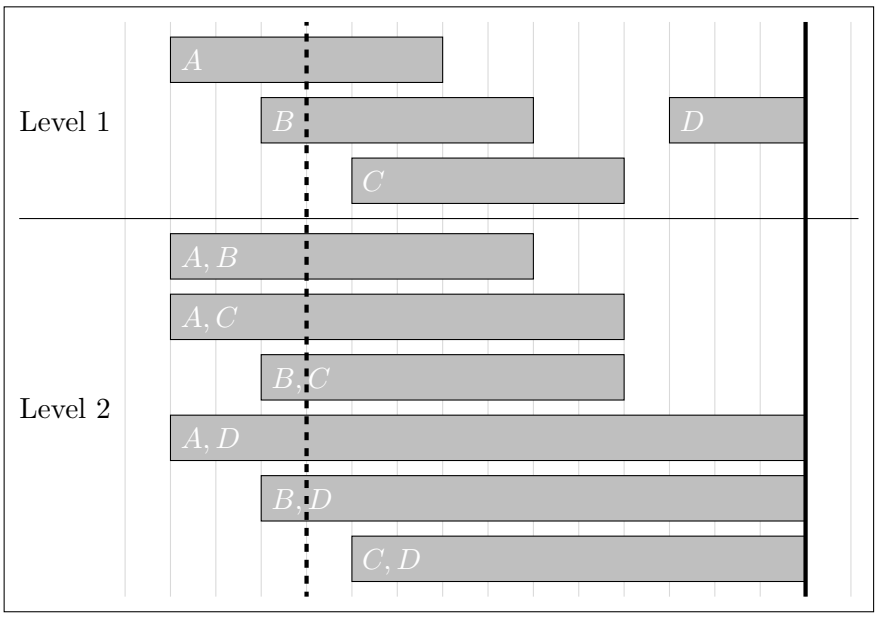

Figure 6.2: A serial association with a three-way overlap. The existence of the serial association increases the number of event types that are created at the second level of event types. The window is marked by the two black vertical lines – solid for the current frame, dashed for the last frame.

6.2 Proxy Ground Truth Comparison

Other than the need to cap the number of levels allowed in the event type hierarchy, the comparison with the proxy ground truth has produced the results least in line with expectations. However, on analysis, there are some signs that these results may be due to flaws in the proxy ground truth, rather than due to flaws in the system's design. In addition, further analysis indicates that even with the flawed proxy ground truth there are some aspects that are in line with expectations. The reason for the proxy ground truth may be flawed is discussed in section 6.3.

By far the largest determining factor for all the comparison measures presented is the model employed. The video duration has some effect but usually stabilises after the first few minutes of input. The noise level has negligible impact. As the hypotheses are linked to the comparison between the models more than any other factor, and are the largest determining factor, the results and discussion shall focus on the comparison between the models. The plots presented will show the duration data as separate lines. This allows the plot to both show the variation that a model can have and show to some extent the influence in the measures that the video duration has. The influence over noise levels will not be shown – each plot is based only on the data for zero noise. This lack of noise data is also justified by the fact that the noise tolerance is analysed separately later on within this chapter.

There is another general influence on the comparison measures presented – the complexity of the learning scenario. Through the three learning scenarios, the complexity affects the measures in two ways: Firstly, there is a reduction in both the typical and peak performance for each measure, which is due to the difficulty of getting a good score against a proxy ground truth with a larger set of rules than a smaller set of rules. Secondly, as the complexity increases the variation in each measure that is based on the video duration increases for each model. This occurs because more complex models take longer for their corresponding observed rule sets to stabilise.

As mentioned previously, some results more in line with expectations can be found within the results. The results more in line with expectations can be separated from those less in line with expectations by their level within the event type hierarchy. The vast majority of the results less in line with expectations can be attributed to the second level of compound event types (i.e. those rules that are compounds of compound event types). This again can be attributed to the much larger proxy ground truth than is present for the second level than the first level in each learning scenario. This is because the second level is derived from the first, which means that any disagreement between the output of the system and the proxy ground truth on the first level will be magnified exponentially on the second level. The exponential divergence arising from the previous multiple overlap problem already discussed.

The first measure that shall be reviewed is the precision measure. Figure 6.3 shows the precision score for each model for both levels of the hierarchy

and figure 6.4 shows the precision score of each model for just the first level of the hierarchy. For some models and scenarios, the precision gives results that are reasonably in line with expectations, and even some that are very in line with expectations, with the Inhibition and Pre-Exposure models performing the best across each scenario. The results least in line with expectations are those of the Temporal / Reacquiring / Blocking group of models, it was expected that these would perform better than the Absolute and Iterative Acquire-Extinguish models.

Of the three non-conditioning models, the performance rank-order for the precision measures is as expected across the models, with the general trend of the Fixed Increment model showing the worst performance, a modest increase in performance for the Symmetrical Fixed Increment model and then a larger increase for the Count Only model. There are some outliers in this trend, but these appear to be for the one minute and two minute inputs, so the rule sets had not stabilised by that point, and the first rules to be generated are more likely to be correct because they reached the significance threshold the fastest. There is another outlier in the precision scores of those three models that cannot be explained by the data points being from the short duration inputs. This is that in figure 6.4c, the Count Only model has a much lower score than the other two models. No explanation was found for this result. Even more anomalous is that in the same subfigure, with the exception of the one minute input, the precision scores for the Count Only model are in the reverse order. Again no explanation could be found for this behaviour.

A similar trend for all of the models with the level one precision results can be seen as that of the all-level precision results, but each point having a generally higher absolute precision value. Defying this general observation is the two Acquire-Extinguish models, which shows some improvement in performance when only taking into account the level one results.

Of all the results presented in this chapter, the recall results are least in line with expectations. However, as with the precision results, the level one results show significant improvement when viewed alone. The recall results for every level are shown in figure 6.5. The corresponding recall results for only the level one event types are shown in figure 6.6. The all-level results show very poor recall performance for every scenario, with the best recall score over every run being 0.042 – effectively meaning that the very best result only found 4.2% of all the proxy ground truth rules. However, taking only the level one results, the recall values are transformed – with the best result being 0.938. This implies that almost the entire issue with these is that there is a great deal of disagreement between the output of the system and the proxy ground truth, and there is little evidence that it is not the proxy ground truth that is being overly broad. As with the precision results, the relative trend between the models for the two sets of plots is broadly the same.

The individual recall results for each of the models appears to be somewhat of a mirror for the precision results – those models that did less well with the

precision measure performed better with recall measure, and those models that performed well with the precision measure did less well with the recall measure. To some extent this is expected, since there is a form of inverse relationship between precision and recall. It is possible to score well with both measures however, so it is not a true inverse relationship. This can be seen through a trivial method of getting a high recall – if every possible rule is generated, the recall would be 100%, as every correct rule would be generated, but this would lead to a very low precision as there would be a very high number of false positives. Conversely, a higher precision can be obtained by producing very few rules, as then each true positive gained would not be divided by a significantly higher number. This appears to be what has happened in the results. The two Fixed Increment models and the Temporal group of models generated the highest absolute number of rules and so created a better recall, but at a high cost of precision. The Count Only, Inhibition and Pre-Exposure models produced considerably fewer rules, giving a higher precision at the expense of recall. The two Acquire-Extinguish models then fell in the middle for the number of rules produced, and so fell in the middle for both precision and recall.

The recall results demonstrate that there is a difference between the Temporal and Reacquisition models showing a deviation in both the all-level and level one results. Over every result in this section, a difference between the results of the two models is rare. The reasons why the Temporal, Reacquisition and Blocking models have such similar results is discussed in section 6.5.

With the balance between precision and recall being different for each model, the results of the F_1 measure and the MCC becomes of even greater interest. The results of both measures take a similar broad shape and so both measures will be discussed simultaneously. The very low recall figures for the data that includes both levels heavily influences the F_1 measure and the MCC which means that their respective all-level data is also less in line with expectations. Because of this, only the plots for the level one data are presented. Figure 6.7 presents the level one data for the F_1 measure and figure 6.8 shows the level one data for the MCC.

The most remarkable observation between the F_1 and MCC measures is how similar both sets of results are – with a few exceptions, the MCC takes almost the exact same shape as the F_1 measure, but at a slightly higher absolute value. The slight increase can be accounted for by the inclusion of the very large true negative values in the MCC calculation. Of the exceptions to the similarity between the two plots, the MCC slightly favours the two Fixed Increment models, the Inhibition model and the Pre-Exposure model and slightly disfavours the Temporal group of models. As the learning scenarios become more complex, again the values of the measures in general reduce. Overall, it appears that the models that demonstrated a higher recall (i.e. the two Fixed Increment models and the Temporal group of models) have produced higher scores in these two measures than the models that demonstrated

a higher precision (i.e. the Count Only, Inhibition and Pre-Exposure models), although the discrepancy between the two groups is certainly less than either the individual precision or recall results. As the recall values are worse than the precision values are good, the end result is that the more indiscriminate models are preferred over the more discriminating ones.

Other performance comparison measures not discussed in section 5.4 or presented in this section were collected and calculated. However, the reason these measures have not been fully included in this thesis is that they are not a discriminator for any of the four input variables. These measures are the specificity and the accuracy. The specificity is a measure of the proportion of true negatives out of all the negatives found, which is calculated using equation 6.1. The accuracy is a measure of the proportion of correct results that were found out of the set of all possible rules, which is calculated using equation 6.2. The reason the measures were not used is because for every value calculated, the answer returned rounded to the value one. This is because the size of all possible results is vast compared to any of the output rule sets or the proxy ground truth rule sets. This means the true negative set size will also be much larger than the other set sizes, causing the result to be very near the value one.

$$\text{Specificity} = \frac{|TN|}{|TN| + |FN|} \tag{6.1}$$

$$\text{Accuracy} = \frac{|TP| + |TN|}{|TP| + |FP| + |TN| + |FN|} \tag{6.2}$$

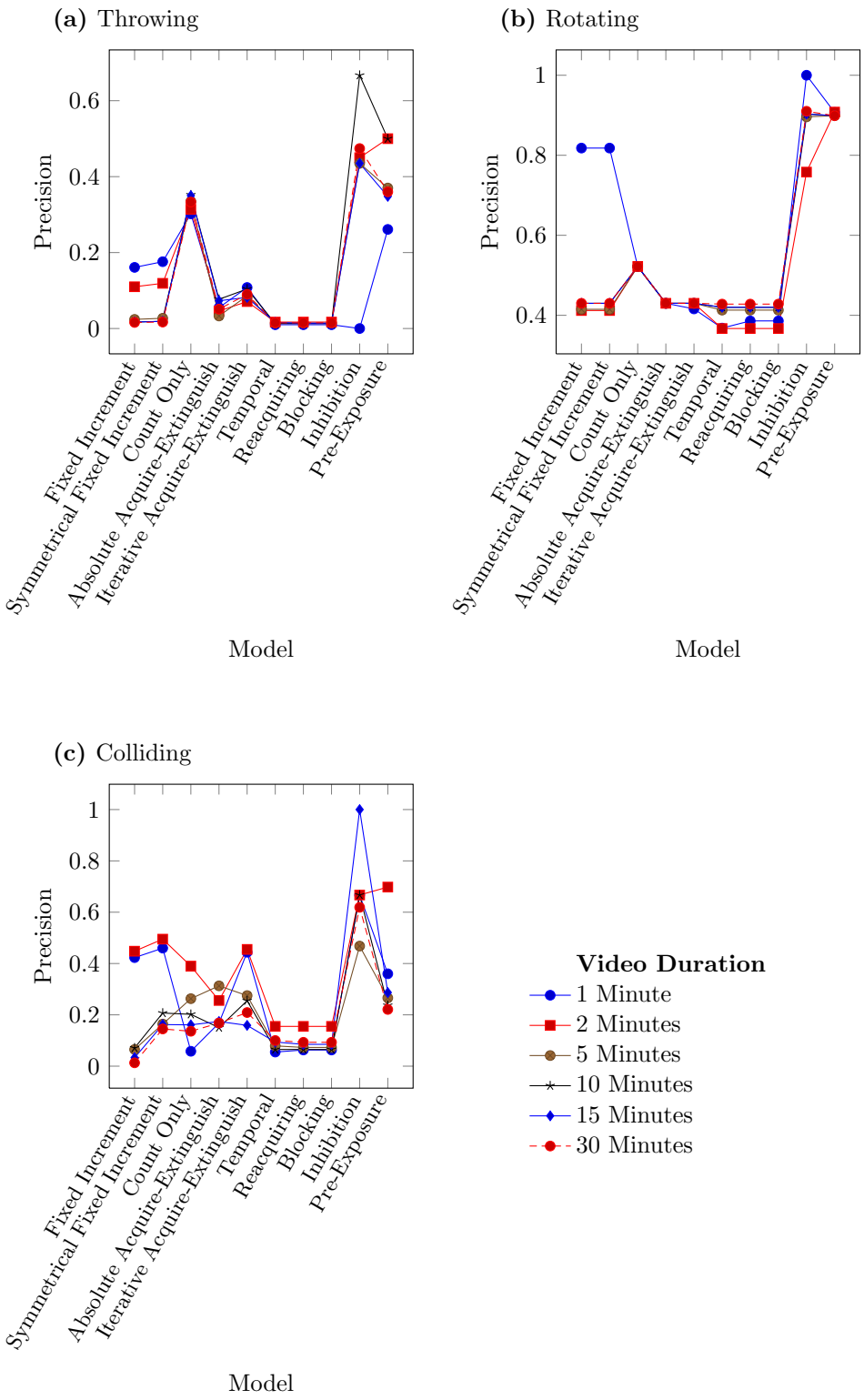

Figure 6.3: A plot comparing the model employed against its precision value for every level of the event-type hierarchy. The sub-figures each show one of the learning scenarios. Each line corresponds to one value of video duration. Only those runs where the noise level was zero are shown.

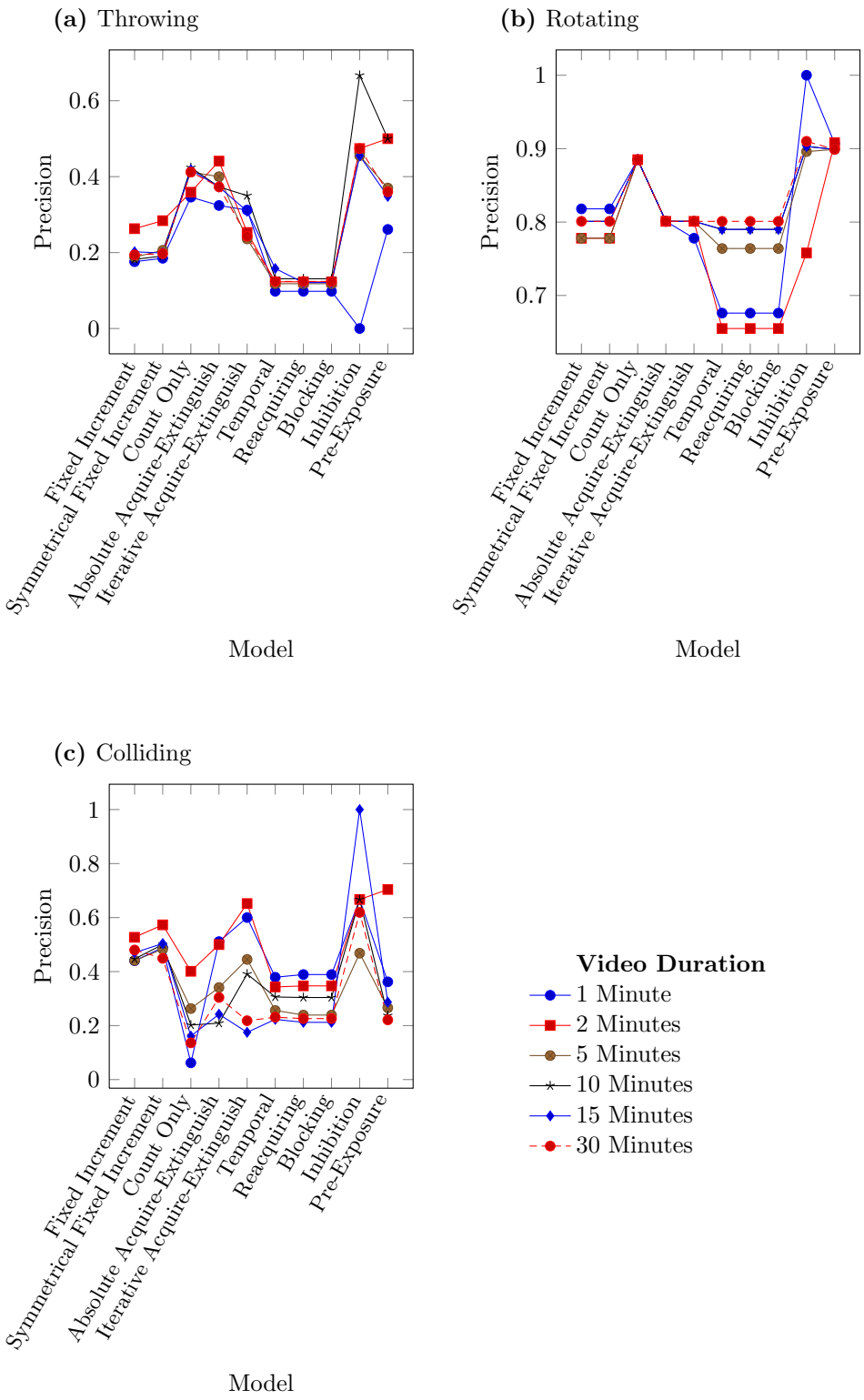

Figure 6.4: A plot comparing the model employed against its precision value for the first level of the event-type hierarchy. The sub-figures each show one of the learning scenarios. Each line corresponds to one value of video duration. Only those runs where the noise level was zero are shown.

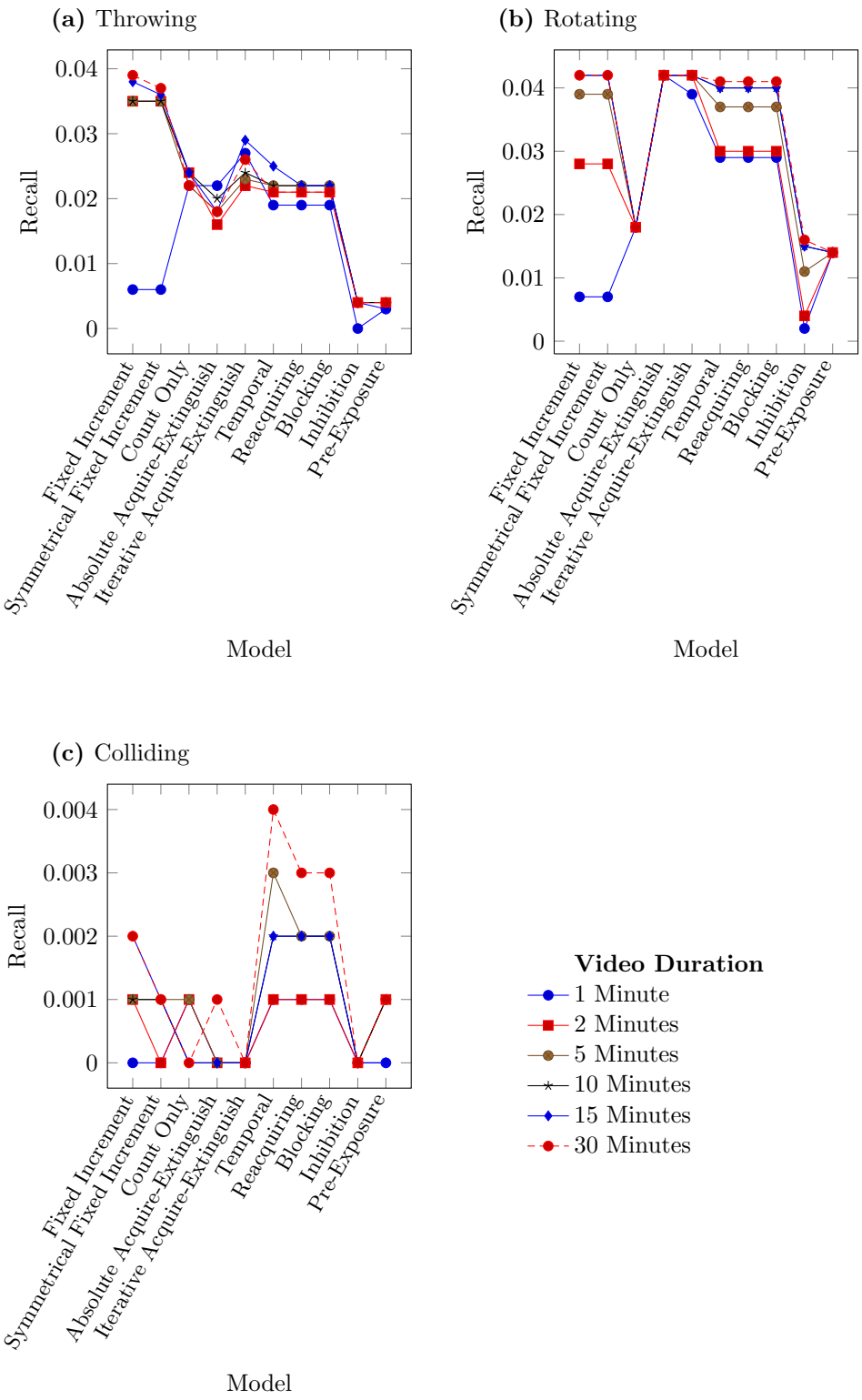

Figure 6.5: A plot comparing the model employed against its recall value for every level of the event-type hierarchy. The sub-figures each show one of the learning scenarios. Each line corresponds to one value of video duration. Only those runs where the noise level was zero are shown.

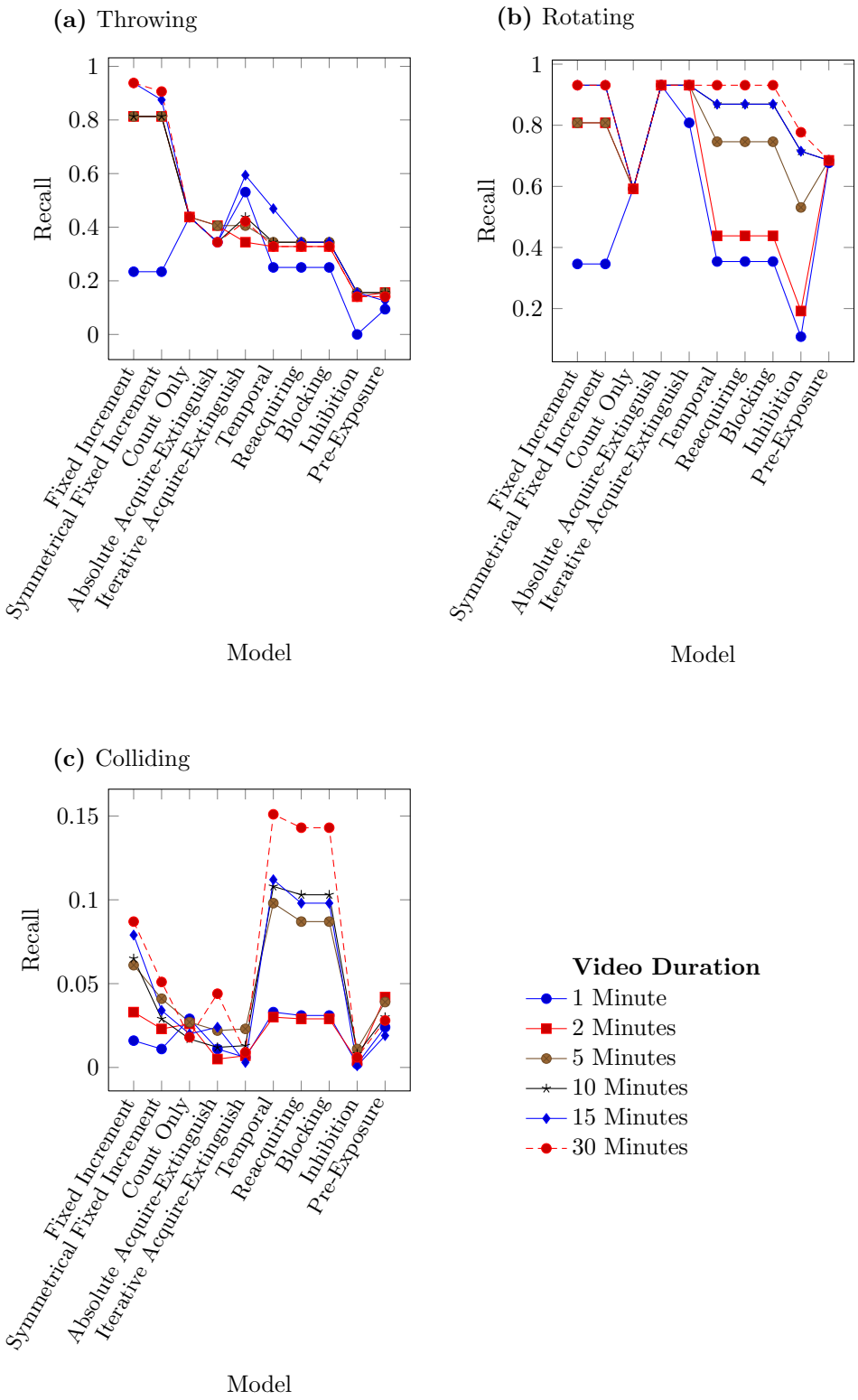

Figure 6.6: A plot comparing the model employed against its recall value for the first level of the event-type hierarchy. The sub-figures each show one of the learning scenarios. Each line corresponds to one value of video duration. Only those runs where the noise level was zero are shown.

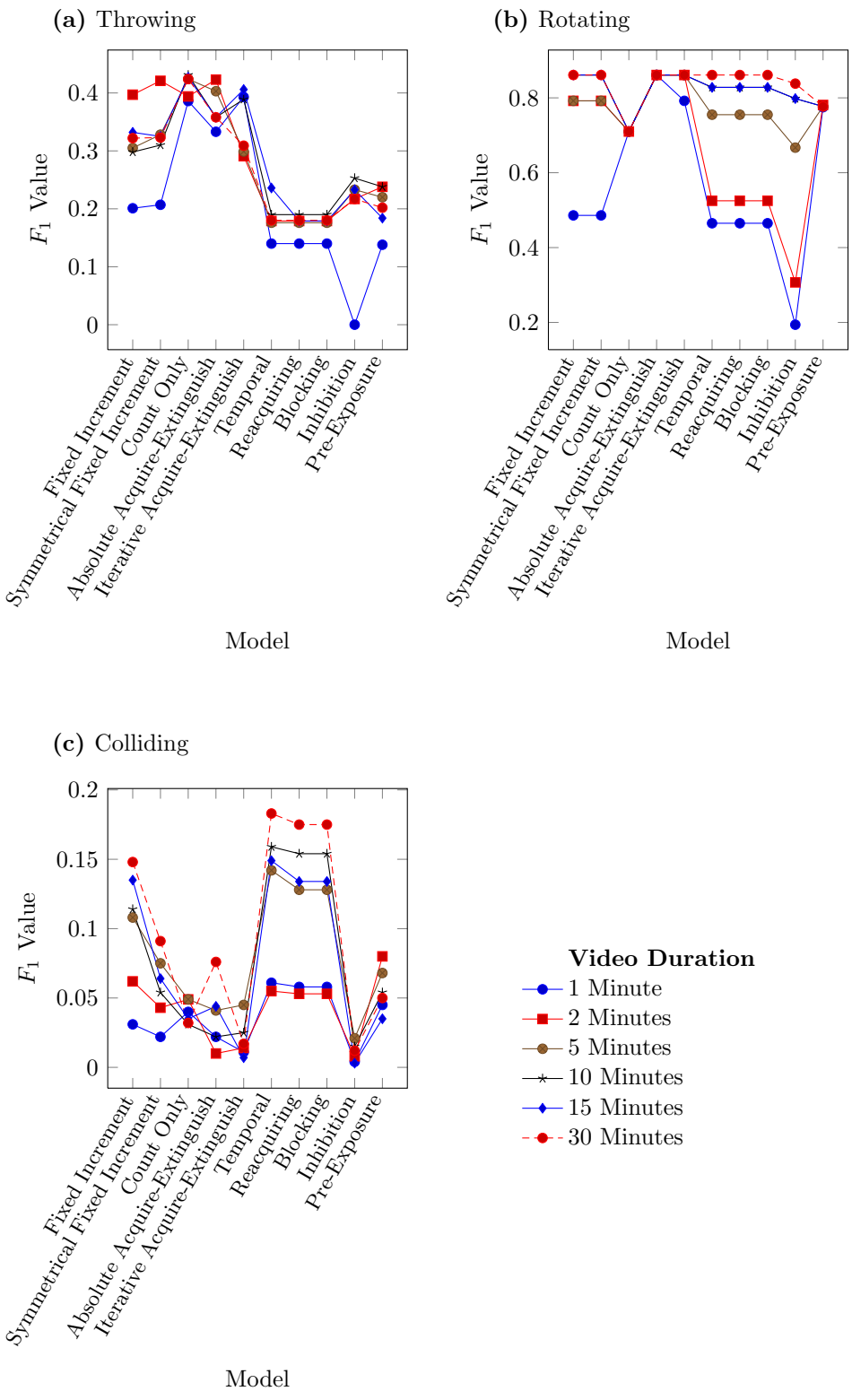

Figure 6.7: A plot comparing the model employed against its F_1 value for the first level of the event-type hierarchy. The sub-figures each show one of the learning scenarios. Each line corresponds to one value of video duration. Only those runs where the noise level was zero are shown.

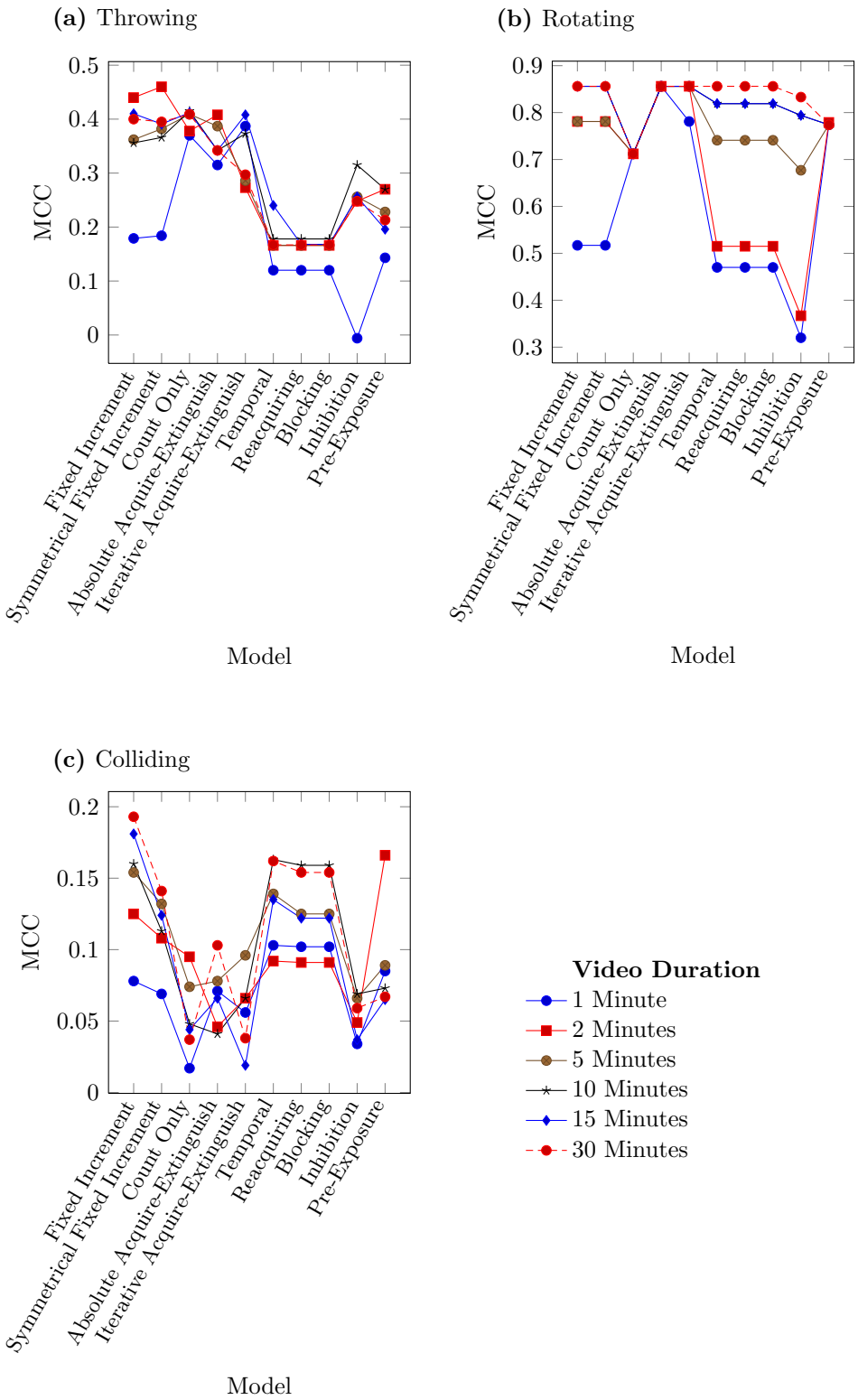

Figure 6.8: A plot comparing the model employed against its Matthews correlation coefficient (MCC) for the first level of the event-type hierarchy. The sub-figures each show one of the learning scenarios. Each line corresponds to one value of video duration. Only those runs where the noise level was zero are shown.

6.3 Qualitative Results

The complete output data set is very large, with output rule sets for 1260 different input combinations and an individual rule set typically having thousands of generated rules. Due to this sheer quantity of data, it is not feasible to conduct a comprehensive qualitative review of these results. This means that any qualitative analysis could easily miss an important detail. However, as long as this risk is acknowledged, a qualitative analysis does provide insight into the operation of the system.

The qualitative results presented in this section focus on the first-level association rules. This is for three reasons. The first reason is that due to the respective sizes of the first and second levels, focusing on the first level rules allows the results presented to be less sporadic. The second reason is that the results from the quantitative comparison against the proxy ground truth suggested that the second-level rules are too inaccurate deal with. The final reason for focusing on the first-level rules is that they are the biggest influence on which second-level rules occur, so a qualitative review of the first level rules at least gives a strong implication of what would be expected to be observed in the second-level rule sets.

In the rules that are presented in this section, the significance value has been left out of the rules for notational clarity. These values present no relevant information to the discussion because by definition all of the rules have a significance value above the upper threshold which means there is very little variance in the values.

There are three approaches that have been taken in reviewing the qualitative results, each is discussed in turn in its own subsection. The first subsection gives the general observations that were made regarding the output of the system. The second subsection then gives a qualitative comparison between the rule sets produced by each of the ten significance models. The final subsection gives an argument the qualitative results suggest that the results that are against expectations in section 6.2 are due to the proxy ground truths not fully representing their respective scenarios.

6.3.1 General Observations

It was observed that the most prevalent and quickly learned associations were those that had some form of logical necessity – in other words, those associations that will always happen due to the definitions of the atomic events. For example, in every output file that was inspected, the rule in equation 6.3 was present, with different object labels.

$$\texttt{happensTogether}\left(\begin{array}{l}\texttt{makeContactRight}\,(O_1, O_2)\,,\\ \texttt{makeContactLeft}\,(O_2, O_1)\end{array}\right) \tag{6.3}$$

This rule states that if the first object makes contact with the second on its right-hand side, then the second object makes contact with the first object

on its left-hand side. Similar rules for the other directions, the break contact events and the merge and emerge events are also almost always present, with the notable exception of the low-noise rotation outputs, which are missing the vertical definitions due to the object only moving horizontally. These rules are nearly always present because the respective atomic event definitions mean that both events are always generated side-by side.

There are some less expected results that involve rules that were learned due to logical necessity that are seen in the models that implement the temporal model's features. These unexpected rules are those that associate two events that always start at the same time, for instance, due to a logical necessity between the events or due to some weaker causative effect. These rules were not expected because if the two events start simultaneously, they will have an inter-stimulus interval of zero because the inter-stimulus interval is defined as the interval between the two start times. An inter-stimulus interval of zero would mean that the models would produce a change in significance value of zero, implying that if two event types always start together, the rule associating them should not appear. One possible explanation for this behaviour is that the event types have associated together from different event instances. Due to the repetitive nature of the input, if the duration of a whole repetition of the scenario was lower than the width of the moving window, an association could form between the two event types that start at the same time.

The next most frequently observed kind of pairing are those associations that are nearly inevitable – in other words, where the case that the association does not apply in is a rare exception. For example, in the throwing and rotating scenarios (and the colliding scenarios that have noise in the input – see section 6.4), every output file that was inspected has the rule shown in equation 6.4.

$$\texttt{happensTogether}\left(\begin{array}{l}\texttt{makeContactLeft}\,(O_1, O_2)\,,\\ \texttt{mergeLeft}\,(O_2, O_1)\end{array}\right) \tag{6.4}$$

Along with the above rule, other similar ones were frequently found such as the counterpart rules for the other three directions and the converse rules associating the emerge and the break contact atomic events. In the association above and others like it, the only time that this association would not be learned is either in the case when objects never make contact, or the case when objects can never pass in front of another object – as in the case of the output rule sets produced from low-noise colliding scenario inputs.

It is almost inevitable that this association occurs in the scenarios it did occur in. This is because, in those scenarios, the need for objects to pass in front of each other is essential to the nature of the scene. In addition, the converse rules involving the emerge atomic events and the break contact were also found in the throwing scenario, associations between the make contact and merge event types were also found for all four of the cardinal directions.

6.3.2 Model Comparison

In order to compare the models qualitatively, a small number of rules were selected from each scenario's proxy ground truth that were deemed to be the most important for describing the scene. The output from the system for the zero-noise input for each scenario was checked to see if they contained the chosen rules. The thirty minute output sets were used as this gives the best chance that the rule sets will have stabilised.

For the throwing scenario, four rules were chosen, which are listed in equations 6.5 to 6.8. The rule in equation 6.5 states that after the ball is no longer in front or behind the person, the ball will be receding from the person. The rule in equation 6.6 states that the ball will move down after it has moved up. The rule in equation 6.7 states that after the ball stops receding from the person, it will approach the person. Finally, the rule in equation 6.8 states that when the ball is approaching the person, it will eventually be in front or behind the person.

$$\texttt{happensTogether}\left(\begin{array}{l}\texttt{emergeBottom}\left(\texttt{Person},\texttt{Ball}\right),\\ \texttt{receding}\left(\texttt{Person},\texttt{Ball}\right)\end{array}\right) \tag{6.5}$$

$$\texttt{happensTogether}\left(\texttt{moveUp}\left(\texttt{Ball}\right),\texttt{moveDown}\left(\texttt{Ball}\right)\right) \tag{6.6}$$

$$\texttt{happensTogether}\left(\begin{array}{l}\texttt{receding}\left(\texttt{Person},\texttt{Ball}\right),\\ \texttt{approaching}\left(\texttt{Person},\texttt{Ball}\right)\end{array}\right) \tag{6.7}$$

$$\texttt{happensTogether}\left(\begin{array}{l}\texttt{approaching}\left(\texttt{Person},\texttt{Ball}\right),\\ \texttt{mergeBottom}\left(\texttt{Person},\texttt{Ball}\right)\end{array}\right) \tag{6.8}$$

In the throwing scenario, the rule in equation 6.5 was observed in all outputs other than the outputs of the Inhibition and Pre-Exposure models. For the rule in equation 6.6, only the Fixed Increment model and the Count Only model produced the rule. Every model produced the rule in equation 6.7 and the rule in equation 6.8 was found in the outputs of all of the models with the exception of the Absolute Acquire-Extinguish model.

For the rotating scenario, seven rules were chosen, which are listed in equations 6.9 to 6.15. The rule in equation 6.9 states that when the two objects are receding from one another, they will soon start approaching each another. The rules in equation 6.10 and equation 6.11 state that after approaching one another, the two objects will merge. The rules in equation 6.12 and equation 6.13 state that after merging with each other on one side, the objects will soon separate from each other on the opposite edge. The final pair of rules, in equation 6.14 and equation 6.15 state that after separating from each other, the two objects will begin receding from each other.

$$\texttt{happensTogether}\left(\texttt{receding}\left(O_1,O_2\right),\texttt{approaching}\left(O_1,O_2\right)\right) \tag{6.9}$$

$$\texttt{happensTogether}\left(\texttt{approaching}\left(O_1,O_2\right),\texttt{mergeRight}\left(O_1,O_2\right)\right) \tag{6.10}$$

$$\texttt{happensTogether}\left(\texttt{approaching}\left(O_1, O_2\right), \texttt{mergeLeft}\left(O_1, O_2\right)\right) \tag{6.11}$$

$$\texttt{happensTogether}\left(\texttt{mergeRight}\left(O_1, O_2\right), \texttt{emergeLeft}\left(O_1, O_2\right)\right) \tag{6.12}$$

$$\texttt{happensTogether}\left(\texttt{mergeLeft}\left(O_1, O_2\right), \texttt{emergeRight}\left(O_1, O_2\right)\right) \tag{6.13}$$

$$\texttt{happensTogether}\left(\texttt{emergeRight}\left(O_1, O_2\right), \texttt{receding}\left(O_1, O_2\right)\right) \tag{6.14}$$

$$\texttt{happensTogether}\left(\texttt{emergeLeft}\left(O_1, O_2\right), \texttt{receding}\left(O_1, O_2\right)\right) \tag{6.15}$$

In the rotating scenario, the rules in equations 6.9, 6.12 and 6.13 were found in the output from every model. The rules in equations 6.10 and 6.11 were produced by every model with the exception of the Count Only Model. The final pair of rules, those in equations 6.14 and 6.15 were found in the output from every model excluding the Count Only, Inhibition and Pre-Exposure models.

In the final scenario, the rotating scenario, a slightly different approach had to be taken. The main descriptive sequence of event types are where two balls approach each other, make contact, break contact and then recede from each other. The issue here is that there are four balls in the scenario, of which any pair of balls is valid. In addition, for making and breaking contact event types, there are four directions in which contact is made and broken. This has meant that to encapsulate the main descriptive sequence, there are 300 rules that would need to be searched for, making a qualitative analysis not feasible. So instead, four rules were searched for that included wildcards to make allowances for the different ball combinations and allowed directions. This is represented in the rules listed below as a star.

There were four wildcard rules searched for in the system outputs for the throwing scenario. These are listed in equations 6.16 to 6.19. The rule listed in equation 6.16 states that when two balls are approaching each other, they will soon be receding from each other. The rule in equation 6.17 states that when two balls are approaching each other, they will soon make contact with each other. The rule listed in equation 6.18 states that when to balls make contact with each other, they will soon break contact with each other. Finally, the rule in equation 6.19 states that when two balls break contact with each other, they will soon recede from each other.

$$\texttt{happensTogether}\left(\begin{array}{l}\texttt{approaching}\left(\texttt{Ball}*, \texttt{Ball}*\right), \\ \texttt{receding}\left(\texttt{Ball}*, \texttt{Ball}*\right)\end{array}\right) \tag{6.16}$$

$$\texttt{happensTogether}\left(\begin{array}{l}\texttt{approaching}\left(\texttt{Ball}*, \texttt{Ball}*\right), \\ \texttt{makeContact}*\left(\texttt{Ball}*, \texttt{Ball}*\right)\end{array}\right) \tag{6.17}$$

$$\texttt{happensTogether}\left(\begin{array}{l}\texttt{makeContact}*\left(\texttt{Ball}*, \texttt{Ball}*\right), \\ \texttt{breakContact}*\left(\texttt{Ball}*, \texttt{Ball}*\right)\end{array}\right) \tag{6.18}$$

$$\texttt{happensTogether}\left(\begin{array}{l}\texttt{breakContact}*\left(\texttt{Ball}*, \texttt{Ball}*\right), \\ \texttt{receding}\left(\texttt{Ball}*, \texttt{Ball}*\right)\end{array}\right) \tag{6.19}$$

In the throwing scenario, all of the models outputted at least one rule that matched the rule in equation 6.16 with the exception of the Absolute Acquire-Extinguish model, the Inhibition model and the Pre-Exposure model. For the rule in equation 6.17, four models outputted at least one matching rule: The Absolute Acquire-Extinguish model, the Temporal model, the Reacquiring model and the Blocking model. All but four models produced at least one rule that matched the rule in equation 6.18, the models that did not being the Fixed Increment model, the Symmetrical Fixed Increment model, the Iterative Acquire-Extinguish model and the Inhibition model. For the final rule, the rule listed in equation 6.19, half the models had output a matching rule. The models that were unable to produce a matching rule were: The Symmetrical Fixed Increment model, the Count Only model, the Iterative Acquire-Extinguish model, the Inhibition model and the Pre-Exposure model.

Table 6.1 summarises the results in this section. Discussion of these results is included as part of the overall discussion that takes place in section 6.6.

		Model									
		Fixed Increment	Symmetrical Fixed Increment	Count Only	Absolute Acquire-Extinguish	Iterative Acquire-Extinguish	Temporal	Reacquiring	Blocking	Inhibition	Pre-Exposure
Rule Equation Number	6.5	✓	✓	✓	✓	✓	✓	✓	✓		
	6.6	✓		✓							
	6.7	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	6.8	✓	✓	✓		✓	✓	✓	✓	✓	✓
	6.9	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	6.10	✓	✓		✓	✓	✓	✓	✓	✓	✓
	6.11	✓	✓		✓	✓	✓	✓	✓	✓	✓
	6.12	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	6.13	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	6.14	✓	✓		✓	✓	✓	✓	✓		
	6.15	✓	✓		✓	✓	✓	✓	✓		
	6.16	✓	✓	✓		✓	✓	✓	✓		
	6.17				✓		✓	✓	✓		
	6.18			✓	✓		✓	✓	✓		✓
	6.19	✓			✓		✓	✓	✓		

Table 6.1: A matrix showing whether a given model was able to match a specified rule. The equation numbers refer the given rules.

6.3.3 Proxy Ground Truth Weakness

The output of the system could in fact be more in line with expectations than the quantitative comparison with the proxy ground truth would suggest. As previously mentioned, the reason for the results that were against expectations in comparing the outputs of the system against the proxy ground truth is believed to be due to each proxy ground truth being a poor representation of its scenario, as opposed to the system's output being a poor representation. The basis of this argument is due to the qualitative review of the results.

In appraising the raw output rules, it was found that some of the false positives are quite reasonable things for the system to have learned. For example, in the throwing scenario, it was found that a large number of the models had produced the rule shown in equation 6.20 and most models had produced the rules shown in equations 6.21 and 6.22. In the throwing scenario, the majority of the models produced the rule shown in equation 6.23 and in the colliding scenario, the rule shown in equation 6.24 was found in all but one of the model outputs.

$$\texttt{happensTogether}\left(\texttt{moveUp}\left(\texttt{Ball}\right), \texttt{moveUp}\left(\texttt{Person}\right)\right) \tag{6.20}$$

$$\texttt{happensTogether}\left(\texttt{moveLeft}\left(\texttt{Ball}\right), \texttt{moveLeft}\left(\texttt{Person}\right)\right) \tag{6.21}$$

$$\texttt{happensTogether}\left(\begin{array}{l}\texttt{moveLeft}\left(\texttt{Ball}\right),\\ \texttt{mergeLeft}\left(\texttt{Ball}, \texttt{Person}\right)\end{array}\right) \tag{6.22}$$

$$\texttt{happensTogether}\left(\begin{array}{l}\texttt{mergeRight}\left(\texttt{ObjectA}, \texttt{ObjectB}\right),\\ \texttt{moveLeft}\left(\texttt{ObjectA}\right)\end{array}\right) \tag{6.23}$$

$$\texttt{happensTogether}\left(\begin{array}{l}\texttt{receding}\left(\texttt{Ball\_0}, \texttt{Ball\_1}\right),\\ \texttt{receding}\left(\texttt{Ball\_1}, \texttt{Ball\_3}\right)\end{array}\right) \tag{6.24}$$

All of these rules in equations 6.20 to 6.24 are reasonable for the system to learn, but were not included in the proxy ground truth. In retrospect this was due to assumptions made about each scenario. If these assumptions were not made, then it would not have been feasible to create the proxy ground truth in the first place, due to the increased complexity.

In the throwing scenario, the rule in equation 6.20 is due to the fact that the when a person throws a ball, their arms move up, as their arms move up, the centre of the person's bounding box moves up. This rule was not included because the relevant proxy ground truth was created assuming that the motion of the person was irrelevant. The rules in equations 6.21 and 6.22 are caused by the simulation of some horizontal movement in the ball and the person tracking the ball, leading to both moving left (or right) at a similar rate and at the catch, the ball will merge on that edge, as well as the usual top edge of the person with the bottom edge of the ball. These two rules were not included because of the previous assumption stated and the assumption that the horizontal motion of the ball was irrelevant.

In the rotating scenario, the rule in equation 6.23 is due to the need to assume the durations of each atomic event instance in making the proxy ground truth. Some of the assumed timings were incorrect, which meant that the two event instances occurred close enough to be associated, but were not close enough with the assumed event instance timings when creating the proxy ground truth.

In the colliding scenario, the rule in equation 6.24 can be seen to be due to there being more than two balls in the scenario, which ball one is moving away from two of the other balls simultaneously. This would be highly likely to occur when there is more than one ball. However, in order to allow for the creation of the proxy ground truth to be tractable, it was assumed that the event type associations only ever involved two balls. The final proxy ground truth then allowed for the fact that there were four balls in the scenario by applying the generated two-ball rule set to each pair of balls in the four ball set.

These qualitative observations suggest there may be another contributory factor for some of the weaker than expected proxy ground truth comparison results. Some of the results could be attributable to the use of a flawed proxy ground truth in each scenario. By extension this suggests a flaw in the methods used to produce each proxy ground truth. It is likely that the way human judgement and deterministic processing were combined was too prescriptive, leading to rule sets for each proxy ground truth that were larger than they should be in some aspects yet smaller than they should be in other aspects.

The qualitative analysis in section 6.3.2 showed a series of rules that were deemed to be those key to describing each scenario. The summary of which models produced a match for each rule in table 6.1 has a far higher match rate than would be suggested by the comparison with the proxy ground truth. This fact that of a better match with some independently selected core rules, again suggests that the proxy ground truth rule sets are a poor representation of their respective scenarios.

There is a final observation regarding the qualitative nature of the system's output that contributes to the argument about the proxy ground truth. This observation is of a more speculative nature, but is still worth expressing. In the not-insubstantial inspections made of the output during the qualitative analysis, no rules were found with the zero-noise outputs that were not explainable in the context of the scenario the rule set came from. However, it has to be again emphasised that as with the rest of this section, no systematic qualitative review of the results could ever have been conducted and so this observation could well be wrong. This being said, even if there are some anomalous rules, they cannot be wide-spread as if that were the case, the inspections conducted would have detected them.

6.4 Noise Robustness

Of the results presented, the noise robustness results are the most in line with expectations. While the system and the models are influenced by noise, they show considerable robustness. For most models the difference between 5% noise and 30% noise is typically less than 10% for each measure considered.

As described in section 5.4, the measurement of noise robustness is entirely independent of the comparison against the ground truth. In addition, as described in section 6.2, it is probable that it is the proxy ground truth that is the cause of the results that were against expectations, rather than the design of the system. The implication of both of these facts combined is that the significance of the noise robustness results is not diminished by the proxy ground truth results that were not in line with expectations.

The results are in line with expectations over both level one and level two result sets. This can be explained by the fact that with the noise robustness figures, the comparison is against the output of the noise-free version of the same input data. By comparing the results with other actual output from the system, there is no possibility of bias within that comparison data. As both the level one and level two data is in line with expectations, the plots presented in this section use the data for every level of the event type hierarchy, rather than just the first level.

All three of the non-noise input parameters – video duration, model selection and scenario complexity – show a comparable level of influence on the system's noise robustness. As the focus for the testing is on the comparison between the models, the effect due to video duration is not displayed in the plots presented; this is because to do so would double the number of plots (one for each measure) but convey little extra information beyond this discussion. The model selection and scenario complexity are represented within the plots however. The values presented in the plots represent the mean noise response over the set of all video duration values.

In each presented plot, there is a recurring and very noticeable result for the colliding scenario. This is the very sudden drop between 0% noise and 5% noise followed by no further response to noise (i.e. the plots for each model roughly remain horizontal – showing little to no influence by the amount of noise). This sudden spike over every plot has a common explanation. In the collision scenario, the objects are not allowed to overlap, as this would go against the point of the scenario. However, when even a little noise is added to the position of an object, there is suddenly scope for overlap states to occur. This in turn introduces an extra set of atomic events which were not a part of the noise-free dataset. This introduction of extra atomic events will in turn create composite events that were not part of the dataset, and so rapidly reducing each robustness measure employed for even low noise levels. To some extent, the throwing scenario also exhibits a faster-than-typical drop in each measure between 0% and 5%, indicating the possibility that an analogous process is happening with that scenario too.

The two Fixed Increment models seem to be some of the most robust to noise with every measure. This may be because those two models had the largest absolute number of rules in the comparison set. This means that any rules that in other models get added through noise are already present in the zero-noise rule set – reducing the potential for there to be a difference in the rule set due to noise. This effect is likely to account for the vast majority of the initial differences between the models, as for the most part those models that produce the fewest rules in their zero-noise rule set are those that have the lowest robustness to increasing levels of noise. Because of this, discussing the absolute value comparison between each model is to some extent meaningless. Instead, the relative change between the models shall be discussed.

The precision of each model's noise response is shown in figure 6.9. After initial reductions are taken into account, the precision data for most models shows only a very slow reduction as the noise is increased. The Count Only model shows the most resilience in retaining precision as the noise increases once each scenario is taken into account – remarkably, for the rotation scenario, the Count Only model shows no average reduction at all. This appears to be an effect of the use of the mean value, as there are some values that do show an effect due to noise, but the majority of the values do not show any or very little effect of noise. The two Acquire-Extinguish models and the Temporal group of models each produce a similar absolute and relative change in noise levels, with these models only being beaten in the lack of relative decline by the Count Only model. Showing very similar relative results but with a higher absolute value, are the two Fixed Increment models, with the higher absolute value attributable to the effect discussed in the previous paragraph. The models most affected by noise in both an absolute and relative manner are the Inhibition and Pre-Exposure models, but even these models, for two of the scenarios, give a reasonable absolute result.

The recall for the noise response for each result is shown in figure 6.10. Again, in general, once the initial reductions have been taken into account, the noise robustness for the majority of the models is reasonably in line with expectations. The most noticeable observation of the recall result set is the very poor relative performance of the Count Only model. This result mirrors the very good performance of the precision data, in a similar way to the mirroring that occurred in the proxy ground truth comparison data. This mirroring can also be seen in the better relative performance of the Inhibition and Pre-Exposure models, which now are as good as the Temporal group of models and show the best relative robustness to noise. The Inhibition and Pre-Exposure models do still have a lower absolute value, but this can be accounted for by the low number of rules produced. Again, the two Fixed Increment models show little decline due to the large starting size of their input.

The results for the F_1 measure and the MCC are plotted on figures 6.11 and 6.12 respectively. The most prominent observation between the two mea-

sures is that they have both produced results that are even more similar than for the proxy ground truth – the largest relative divergence in value between the two plots is so small that it cannot be seen on the plots, even when the plots are overlaid on top of one another. There is some absolute difference, which can be observed in the slightly different scale used on the y axis for the two measures, with marginally better results being given by the MCC.

With all the measures of noise robustness presented in this section, it is the rate of decline in performance as the noise increases that is a better measure of how resilient a model is to noise more than the absolute performance. This is because a low rate of decline indicates ability for a model to maintain consistency, which is the desirable attribute for resilience to noise. Because a steep rate of decline indicates a lower robustness to noise, the Count Only model shows by a large margin the least robustness to noise for the F_1 and MCC measures.

The greatest relative robustness is demonstrated by the Temporal group of models, presumably because of consistently good relative noise robustness. In the same manner, due to consistency and the effects previously described, the two Fixed Increment models are the second best for relative robustness and best for absolute robustness. The results for the Inhibition and Pre-Exposure models have in general the worst absolute robustness and the second worst relative robustness. It is worth noting though that for the throwing and rotating scenarios, even the Inhibition and Pre-Exposure models have overall proven relatively robust, with the exception of the Count Only model.

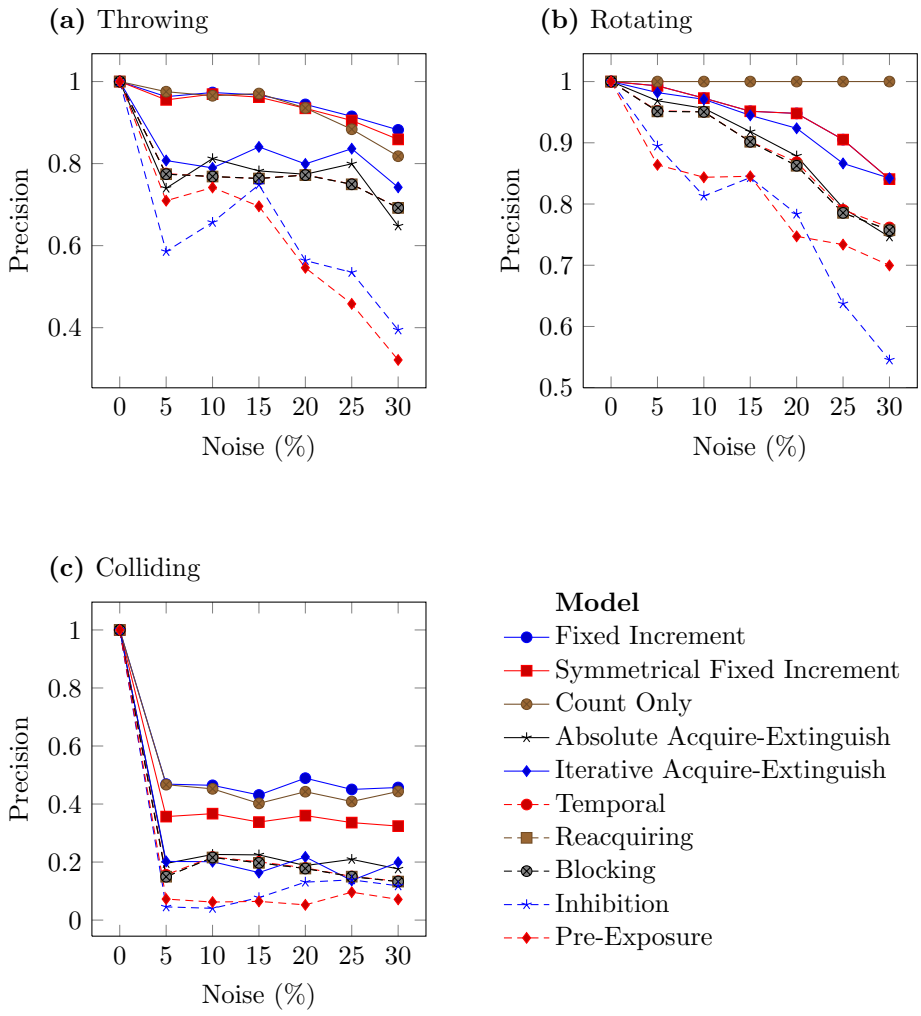

Figure 6.9: A plot comparing the level of noise in the input with the output's precision value. The sub-figures each show one of the learning scenarios. Each line corresponds to one model. Each value shown is the mean value for the output for every video duration value.

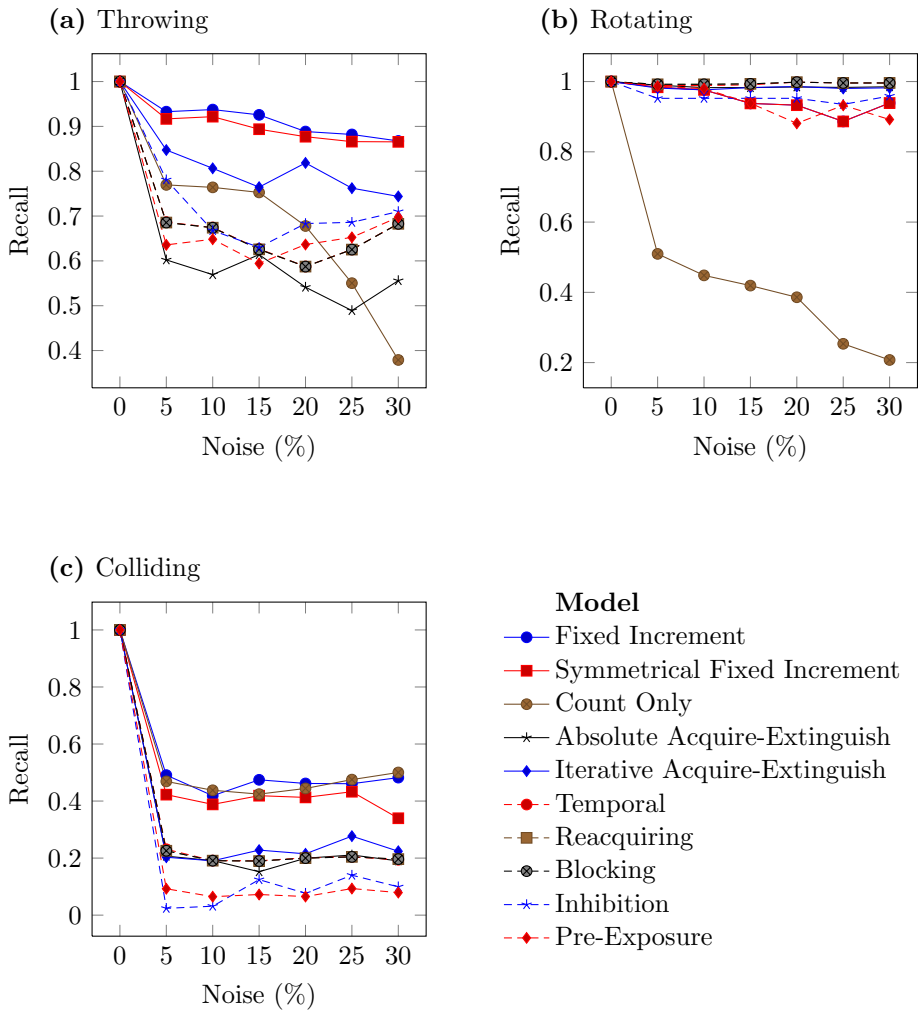

Figure 6.10: A plot comparing the level of noise in the input with the output's recall value. The sub-figures each show one of the learning scenarios. Each line corresponds to one model. Each value shown is the mean value for the output for every video duration value.

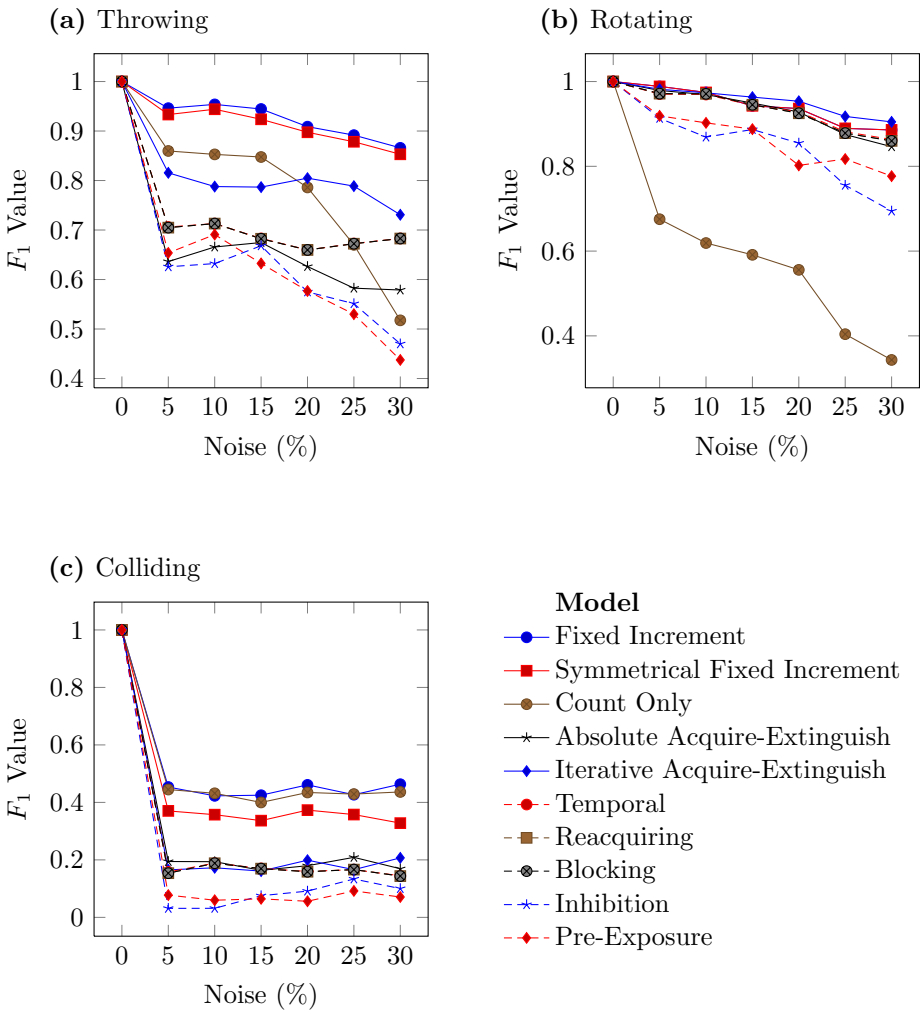

Figure 6.11: A plot comparing the level of noise in the input with the output's F_1 measure. The sub-figures each show one of the learning scenarios. Each line corresponds to one model. Each value shown is the mean value for the output for every video duration value.

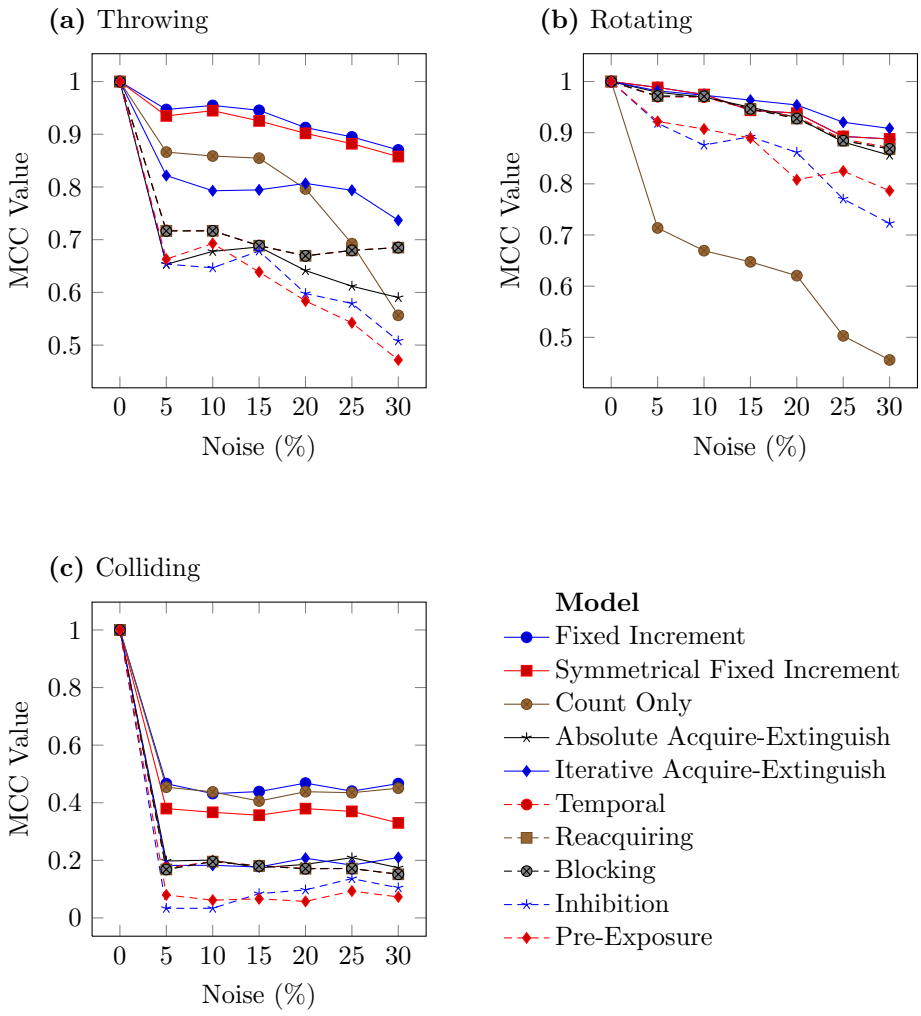

Figure 6.12: A plot comparing the level of noise in the input with the output's Matthews correlation coefficient (MCC). The sub-figures each show one of the learning scenarios. Each line corresponds to one model. Each value shown is the mean value for the output for every video duration value.

6.5 Computational Performance

The processing time performance was measured for each separate input, based on the number of rules the system outputs after processing and the time taken to do so, as described in chapter five. Unsurprisingly, it appears that the main factor determining the mean frames processed per second is the number of compound event types, as measured by the number of rules the system outputs. This is unsurprising because each rule needs to be checked to see if it has been matched once per frame. A scatter plot comparing the two variables for every output instance at every noise level of each of the three scenarios is shown in figure 6.13. The figure shows an inverse correlation between the mean frames per second and the number of rules, which reduces rapidly before tailing off asymptotically towards zero. The gap of results and subsequent increase in the frame rate between counts of 200 and 600 rules in sub-figure b is believed to be due to the fact that the distribution of the video duration times is not linear, and that this is reflected in the observed gaps.

In turn, the main factor determining the number of rules produced is the video duration. As the video duration increases, the number of rules also increases; in general this association follows a pattern of rapidly increasing for smaller video durations but slowing for longer durations. This pattern is shown in figure 6.14, which demonstrates how the number of rules produced responds to differing durations for video. The pattern of rapid growth followed by slower growth mirrors the drop in frame rate shown in figure 6.13. As it is the main factor determining the number of rules, the length of the video in turn is indirectly but strongly related to the mean frame rate. This indirect association is shown in figure 6.15. It is unsurprising that a longer video duration produces a larger number of rules, as a longer video duration gives greater opportunity for all rules to reach the significance threshold.

There are other factors that influence the number of rules created. Comparing between the sub-figures of figure 6.14, it can be seen that the number of rules is also related to the scenario being learned. This can be seen in the different scales of each plot that more rules are created as the complexity of the scenario increases. In these scenarios the complexity can be seen in the number of objects in the scene and the variability of the motion being learned.

The model being employed also affects the number of rules that are created, as can be seen in figure 6.16. The Fixed Increment model shows the highest number of rules, which is to be expected given that the model does not implement any way in which a significance measure can be reduced, meaning that any event type pair will become a rule no matter how uncorrelated if through happenstance it appears enough times.

The Symmetrical Fixed Increment model, which introduces a rule removal mechanism, has results that vary between no removals in the number of rules to a near complete removal. This variation can be explained by the complexity of the scenario – in a complex scenario where there is more variation in the movement and so a greater freedom for pairings to occur due to happenstance.

The Count Only model has some of the lowest rule counts, which can be explained to be due to the need for both congruity and full contingency, a requirement also shown in the Pre-Exposure model, which is comparable in magnitude for the numbers of rules produced. The Count Only model has the interesting artefact that except for the most complex scenario, every result in a scenario has the same number of rules – this suggests that the model converges extremely quickly for simple scenarios, and can be explained at least partially by the fact that a rule can have a significance value above the threshold from the very first observation of a rule.

There is some variance between the Absolute and Iterative Acquire-Extinguish models, which was unexpected given their common basis. Upon investigation it appears that the result is due to the fact that once the significance value in the Absolute Acquire-Extinguish model has been reduced through extinction, that quantity cannot be regained. To understand why the Absolute Acquire-Extinguish model cannot regain any loss in the significance value, the significance value update function for the Absolute Acquire-Extinguish model is repeated in equation 6.25.

$$V_n = \frac{1}{1 + e^{-k_1\epsilon^+}} - k_2\epsilon^- \tag{6.25}$$

With the absolute equation, the positive and negative evidence counts (ϵ^+ and ϵ^- respectively) are counted separately. Consider the case where an event type has reached the plateaux of the sigmoid curve for the positive evidence (the first term in equation 6.25). If some negative evidence for the event type is then received, the linear subtraction of the second term would reduce the significance value as expected. However, this subtraction cannot be reversed by further positive evidence as the first term has already reached its plateaux. The same is not true for the iterative function as there is a singular internal state that both positive and negative evidence contributes to, rather than the two separate internal states of the Absolute Acquire-Extinguish model.

The most counter-intuitive result is that for two of the scenarios, the increase in the number of results in the Temporal model from the Iterative Acquire-Extinguish model. The Temporal model expands on the Iterative Acquire-Extinguish model by adding a curve that modifies the gain in significance value from positive evidence to account for the timing effects. Crucially, that curve always acts to reduce the gain in significance value from a piece of positive evidence. It would thus be logical to assume that this would cause a reduction in the final rule count. This is because it would take more event instances to get an event type past the upper significance threshold and so fewer event types can be expected to reach the upper significance threshold within the duration of the input. Instead, as shown in figure 6.16, for the throwing and colliding scenarios, the rule count quadruples in some cases. An explanation for this result has not yet been found.

The fact that the Reacquiring and Blocking models have nearly the same results as the Temporal model is more explainable, however. The similar results between the Temporal and Reacquiring models can be accounted for by the weak effect of reacquisition. This is because the reacquisition effect is both subtle and for it to be truly visible, requires that associations are extinguished and reacquired many times. The fact that the results between the Reacquiring and Blocking models are almost identical on every input can be explained to be due to the fact that none of the scenarios selected have any sequences that would cause the blocking phenomenon to be largely applicable. These two effects can be observed throughout all the results, with either no or a small change in results between Temporal and Reacquiring model and then no change in results between the Reacquiring model and the Blocking model.

The reduction in the number of results for the models that include conditioned inhibition is consistent with conditioned inhibition being a phenomenon that acts to reduce gains in significance value. Less consistent however is the slight gain in the number of rules as the pre-exposure mechanism is introduced – the mechanism is again one that should act to reduce the gain in association strength, as with the gains in the Temporal model, this has no obvious explanation based on the function of the model, but unlike the Temporal model, the gains are slight enough that it could be down to chance.

Figure 6.17 shows a plot of the frames per second for each of the models. This mostly follows a rough inverse of the corresponding rule counts, although there are a couple of results that do not follow this pattern. The first notable result is that the Inhibition model appears to out-pace the other models for every video duration value, some by a very large margin. This can be explained to be due to the non-linear gains in frame rate for reductions in the number of rules produced. A second notable result is that this plot is one place where the results for the Blocking model do differ from the Reacquiring model. The explanation for this is that the Blocking model still has to check for blocking event instances, even if there are none present. The final notable result is that the Count Only model, despite producing rule counts that are a similar order of magnitude as the Inhibition and Pre-Exposure models, does not have similar frame rates. This is because each time a piece of positive or negative evidence is observed, it requires three look-ups of event types to access the counts for each: the composite type and the two component types. This is not the case for the Inhibition model which only needs the composite type record, however it is again the case for the Pre-Exposure model, which could account the large drop in frame rate.

The final potential influencing factor in the number of rules produced and therefore the frame rate of processing, the level of noise, has very little effect on either. Figure 6.18 shows the effect of noise on the number of rules produced and figure 6.19 shows the effect of noise on the frame rate. The vast majority of the plots are broadly flat, indicating only a little influence. This is a good indication of the system's overall noise handling capabilities. As always, there

are some exceptions. The largest influence on whether noise plays a role in the number of rules and frame rate appears to be the complexity of the scenario. This is exemplified in the colliding scenario, where the colliding scenario plots are the most chaotic for both measures. It appears that the frame rate of the Inhibition model is the most affected by noise, but it could be argued that this is because it is a larger absolute value, so any relative effect the noise has would be amplified by the larger absolute value. The Temporal / Reacquiring / Blocking group of models appear to be influenced by noise, this time made manifest more in the number of rules produced than the frame rate. The general down-then-up pattern seen in figures 6.18a and 6.18c could be explained through the noise causing the timing of different overlap transition event instances to change, thus reducing the speed at which rules reach the upper significance threshold. At some point, the increase in noise starts adding many more transition event instances, which become recorded as rules, thus increasing the number of rules again. The final point worth noting is the sudden large increase in frame rate for the largest level of noise in figure 6.19c. The best explanation found for this result is that as the motion of the balls and the noise is randomly generated, the particular input video instance used for input for those data points included noise and motion that meant that the noise had less effect on the atomic events generated than would be so for the average case.

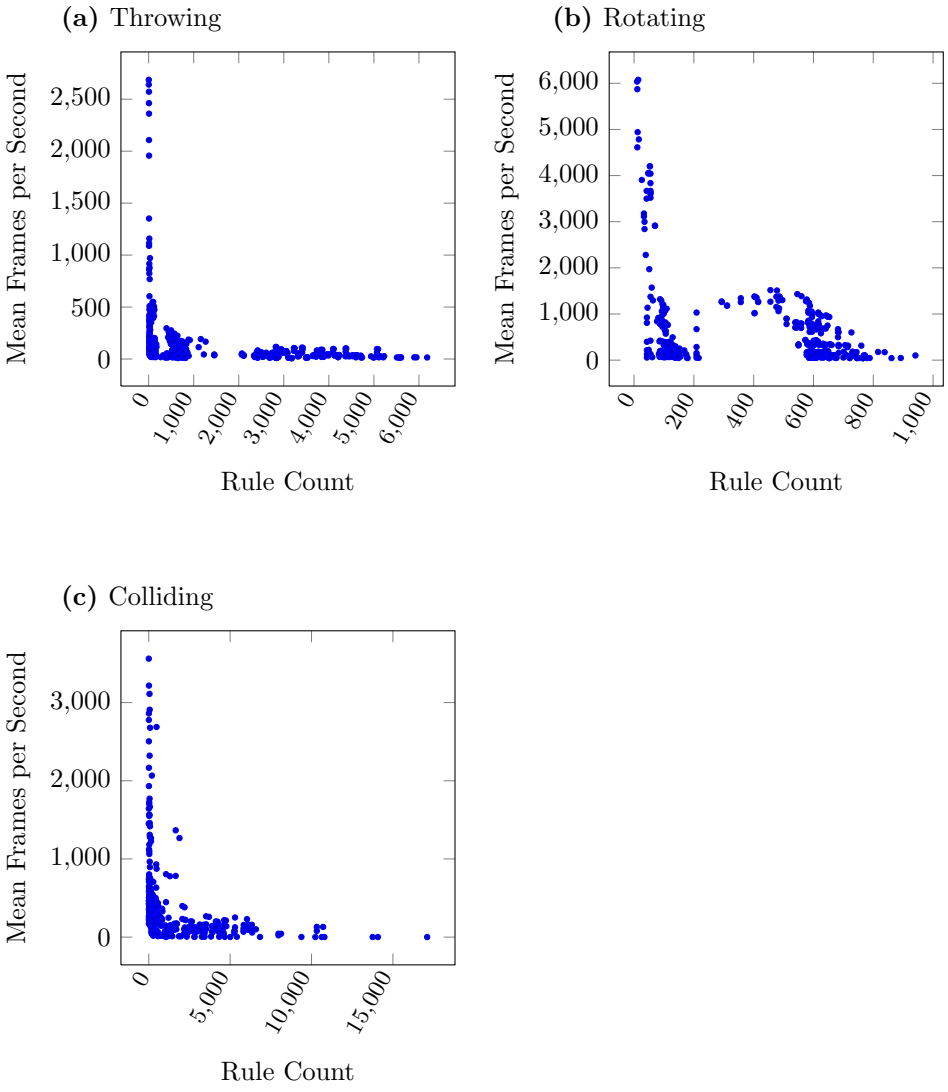

Figure 6.13: A scatter plot comparing the number of rules produced with the mean frame rate for each input provided to the system. The sub-figures each show one of the learning scenarios. All models, durations and noise levels are included.

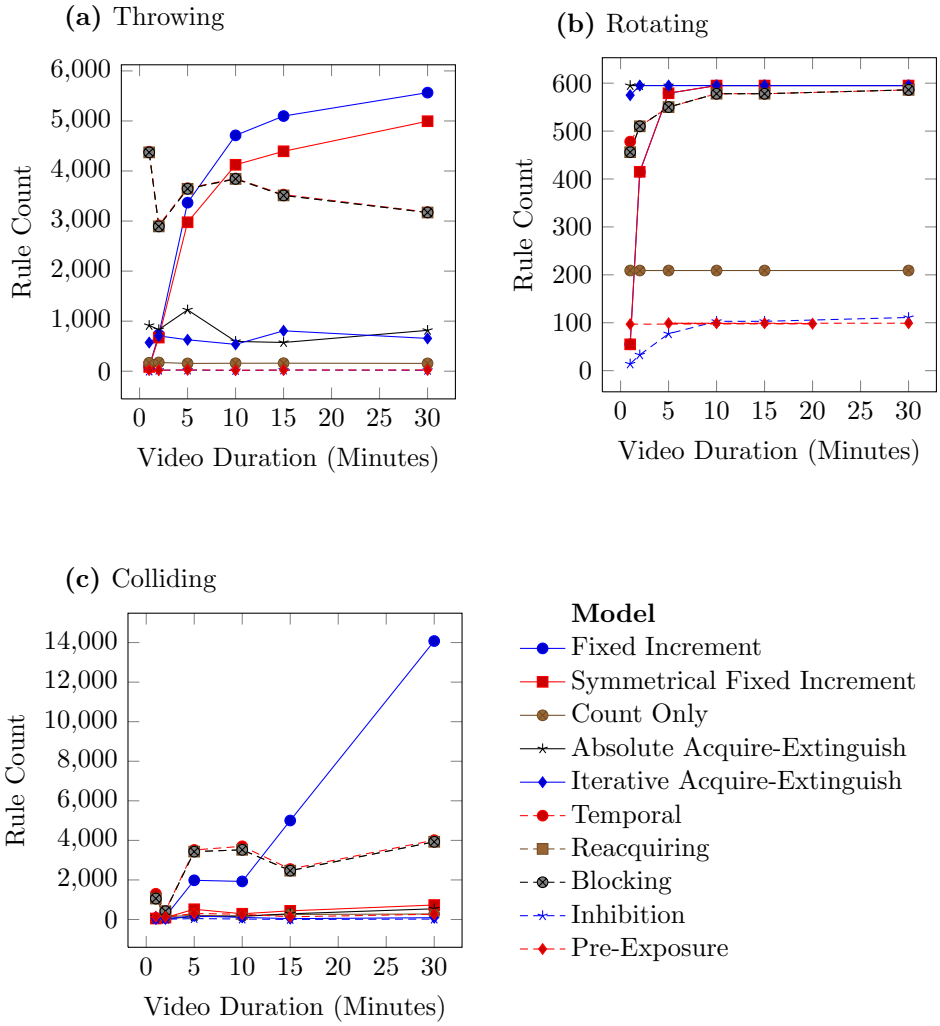

Figure 6.14: A plot comparing the duration of the input video with the number of rules produced. The sub-figures each show one of the learning scenarios. Each line corresponds to one model. Only those runs where the noise level was zero are shown.

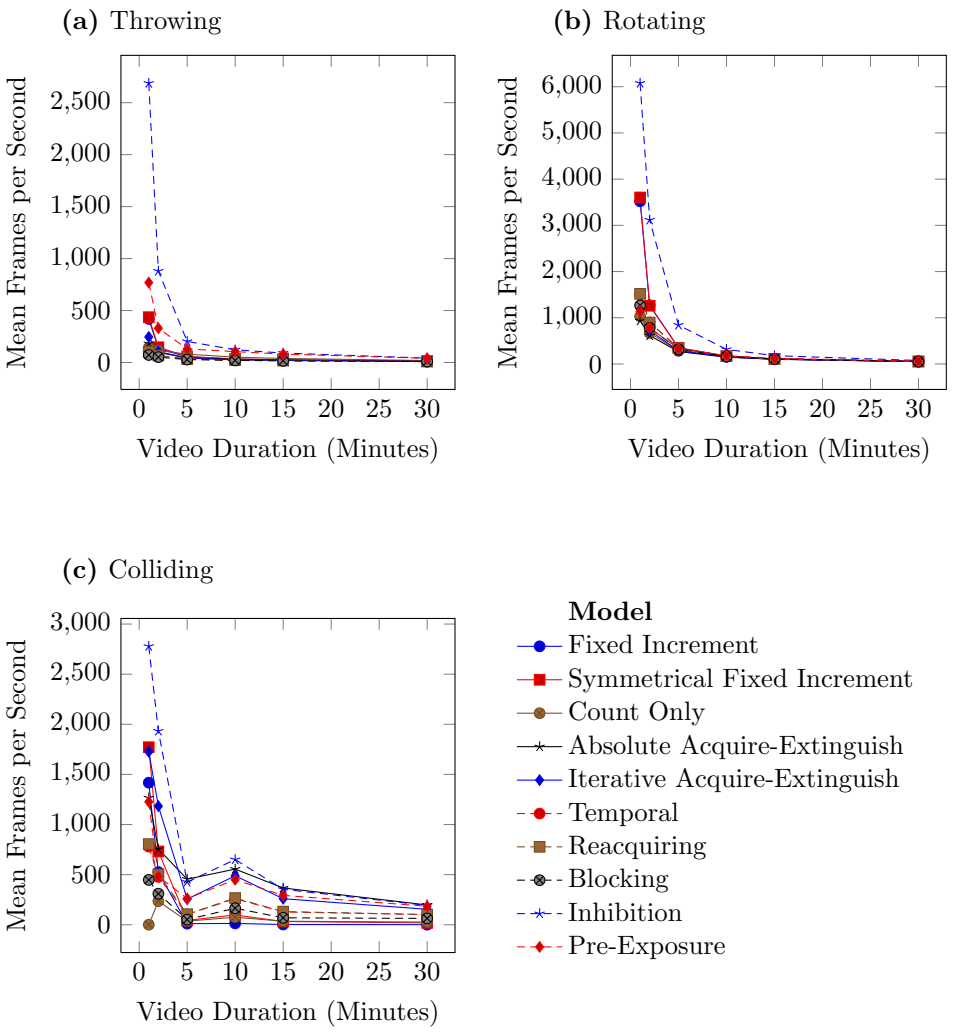

Figure 6.15: A plot comparing the duration of the input video with the mean frame rate. The sub-figures each show one of the learning scenarios. Each line corresponds to one model. Only those runs where the noise level was zero are shown.

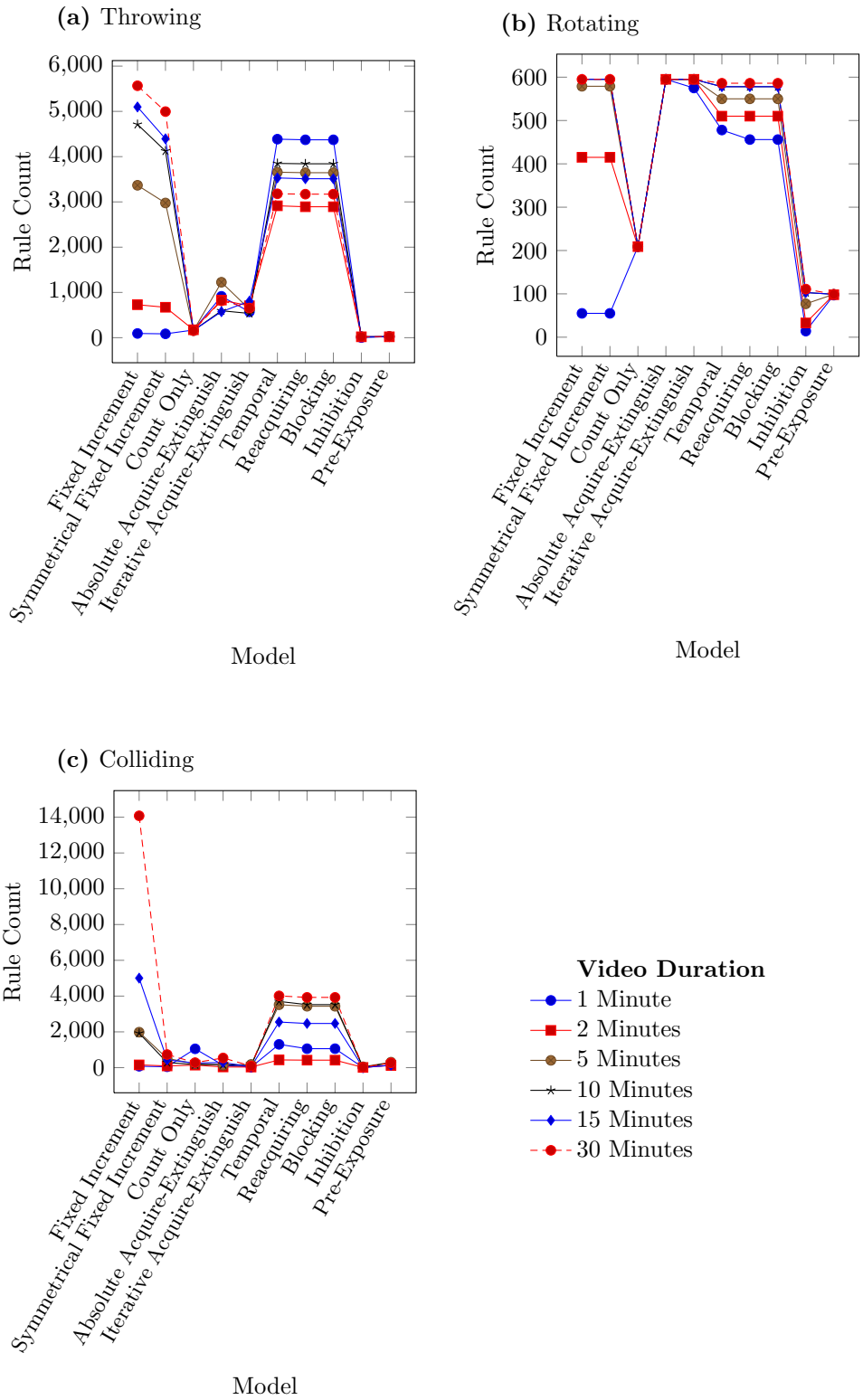

Figure 6.16: A plot comparing the model employed with number of rules produced. The sub-figures each show one of the learning scenarios. Each line corresponds to one value of video duration. Only those runs where the noise level was zero are shown.

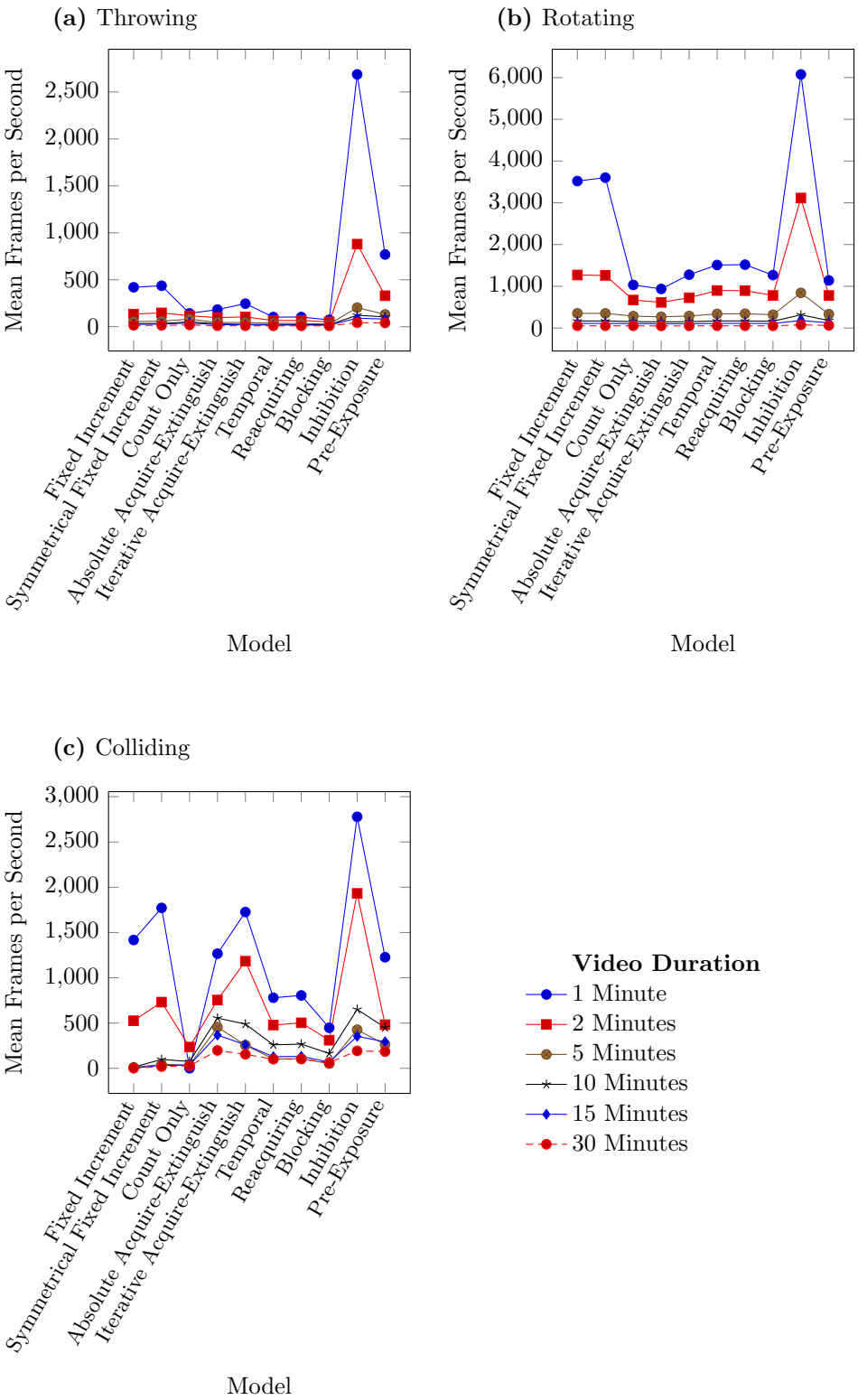

Figure 6.17: A plot comparing the model employed with the mean frame rate. The sub-figures each show one of the learning scenarios. Each line corresponds to one value of video duration. Only those runs where the noise level was zero are shown.

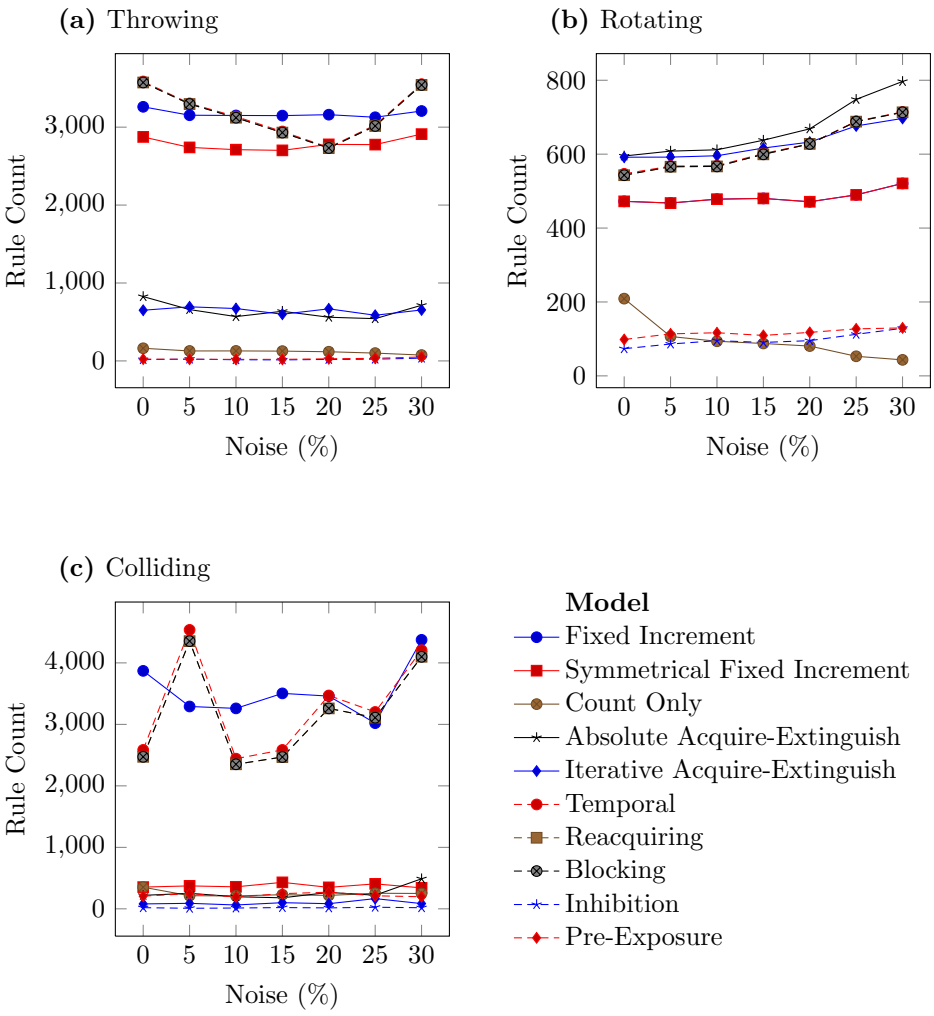

Figure 6.18: A plot comparing the level of noise in the input video with the number of rules produced. The sub-figures each show one of the learning scenarios. Each line corresponds to one model. Only those runs where the video duration is 30 minutes are shown – this level was chosen as it gives the maximum amount of time for the rules to stabilise.

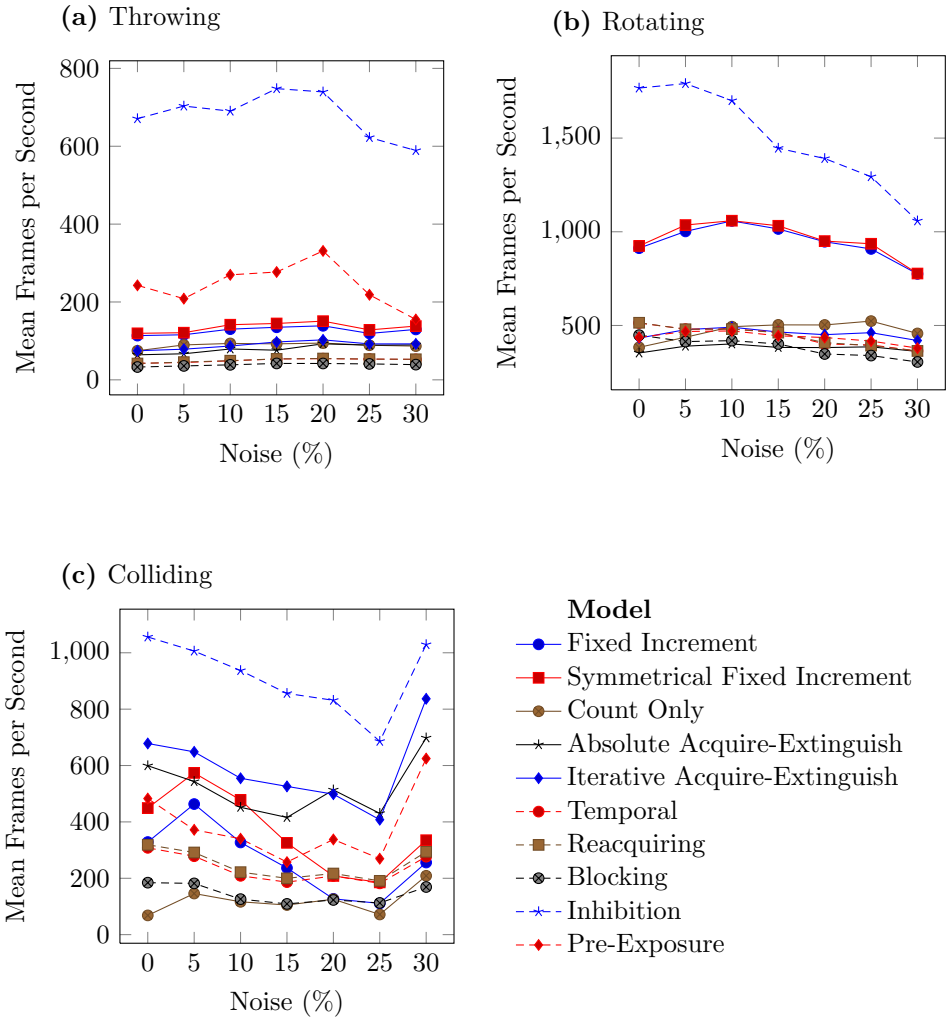

Figure 6.19: A plot comparing the level of noise in the input video with the mean frame rate. The sub-figures each show one of the learning scenarios. Each line corresponds to one model. Only those runs where the video duration is 30 minutes are shown.

6.6 Chapter Conclusion

As mentioned in the introduction to this chapter, the results presented are a mix of both results that are and are not in line with expectations. The result least in line with expectations is that the concept of the event type hierarchy as described in this thesis has been shown to have major problems, both with the combinatorial explosion discovered and the poor proxy ground truth results for the second-level rules. The results most in line with expectations can be found in how robust the system and the models are to the introduction of noise to the input data. In addition the results found for the first-level comparison with the proxy ground truth are somewhat in line with expectations.

In terms of an overall comparison between the models, different models have different strengths. Some models rank well quantitatively, some qualitatively, some are more robust to noise and some are more computationally efficient. This means that producing an overall performance measure to compare the models depends on the importance of each measure. Also, it is not wholly clear-cut how to rank each model within a measure. In order to produce an overall list of how the models compare, a ranking for each individual measure is given with a description for how that ranking was arrived at. These rankings are then combined into a single overall ranking. One of the ways the hypotheses are tested is by seeing if those significance models that implement a greater number of phenomena are also the better all-round models in producing knowledge of a scenario.

The model ranking of proxy ground truth comparison is given in Table 6.2. These rankings are based on the first-level MCC values, shown in figure 6.8. The reason for using this as the basis is that the MCC is considered by this thesis to be a more robust measure of the match between two than the F_1 measure. The ranking considered the mean value for each model, weighted in favour of longer durations the standard deviation was also taken into account. Models were considered tied if the differences in these measures were considered to be negligible.

Rank	**Model**
1	Absolute Acquire-Extinguish
2	Iterative Acquire-Extinguish
3	Pre-Exposure
Joint 4	Fixed Increment
	Symmetrical Fixed Increment
Joint 6	Temporal
	Reacquiring
	Blocking
9	Inhibition
10	Count Only

Table 6.2: The significance model rankings for the results of the proxy ground truth comparison.

Table 6.3 shows the ranking for the qualitative comparison. This was the most straightforward and objective to calculate of all four performance measures. The significance models are ranked by the number of matches against the selected key rules that a model had in table 6.1.

Rank	**Model**
Joint 1	Temporal
	Reacquiring
	Blocking
4	Fixed Increment
5	Absolute Acquire-Extinguish
Joint 6	Iterative Acquire-Extinguish
	Symmetrical Fixed Increment
8	Count Only
9	Pre-Exposure
10	Inhibition

Table 6.3: The significance model rankings for the results of the qualitative comparison.

The ranking for the noise robustness comparison is shown in table 6.4. As with the proxy ground truth comparison, the MCC value shown in figure 6.12 was used for the performance metric. Unlike the proxy ground truth comparison, the mean and standard deviation of each model was not used. Instead the mean gradient was used to rank the significance models. The reason for this difference is that with noise, some degradation of the results is expected, what matters is how the model responds as the noise increases.

Rank	**Model**
Joint 1	Temporal
	Reacquiring
	Blocking
Joint 4	Fixed Increment
	Symmetrical Fixed Increment
6	Absolute Acquire-Extinguish
7	Iterative Acquire-Extinguish
Joint 8	Pre-Exposure
	Inhibition
10	Count Only

Table 6.4: The significance model rankings for the results of the noise robustness comparison.

For the ranking of the computational efficiency comparison, the mean frames per second, as shown in figure 6.17, was chosen to be the comparison metric. As with the proxy ground truth comparison, an input-duration weighted mean and the standard deviation were used to compare the significance models.

Rank	Model
1	Inhibition
2	Pre-Exposure
3	Symmetrical Fixed Increment
4	Fixed Increment
5	Iterative Acquire-Extinguish
6	Absolute Acquire-Extinguish
Joint 7	Temporal
	Reacquiring
9	Blocking
10	Count Only

Table 6.5: The significance model rankings for the results of the computational efficiency comparison.

The final ranking is based on an award of points for each ranking: ten points for being ranked first, nine for being ranked second and so on with one point for being ranked tenth. The points for each were then added, with double weight for the proxy ground truth comparison and the qualitative comparison. The models were ranked in order of how many points were scored over all the other rankings. The overall ranking of the significance models is shown in table 6.6.

Rank	Model
Joint 1	Temporal
	Reacquiring
Joint 3	Blocking
	Absolute Acquire-Extinguish
	Fixed Increment
6	Symmetrical Fixed Increment
7	Iterative Acquire-Extinguish
8	Pre-Exposure
9	Inhibition
10	Count Only

Table 6.6: The overall ranking of the significance models.

In the next and final chapter, chapter seven, the thesis is concluded starting with a discussion of the implications of these results in terms of the hypotheses proposed in chapter one.

Chapter 7

Conclusions

This chapter has three sections, each concluding an aspect of this thesis. The first section looks at the hypotheses presented in chapter one and, incorporating the results and discussion presented in chapter six, discusses whether each of the two hypotheses should be allowed to stand, or be declared to have been falsified. The second section then states the claims of a contribution to knowledge made by this thesis, which are then discussed in turn, looking at whether the claims are supported. The final section of the chapter then looks to what further work could be done using this thesis as a starting point.

7.1 Hypotheses

To recap, the two hypotheses that this thesis proposed in chapter one are re-stated below. This section reviews each of these hypotheses in turn.

Hypothesis A: The phenomena of classical conditioning can be used as a mechanism-independent specification for a system that allows an agent to learn a commonsense knowledge model of its environment.

Hypothesis B: An agent using the phenomena of classical conditioning that passively observes a dynamic environment will still be able to learn a partial commonsense knowledge model of that environment.

7.1.1 Hypothesis A

There was a presumption that as the fidelity of the model of classical conditioning increased, so would the fidelity of the model of the environment learned. This was based on the concept that if the model of classical conditioning increased in fidelity, then there would be a greater constraint on the sort of rules allowed, thus the rules that still passed each of the constraints would be a better match to the optimum model for the environment presented. If this

presumption had held, it would have confirmed the hypothesis by showing a positive correlation between the fidelity of the model of classical conditioning and how well that model fared in the various measures used.

This presumption has not held true in the results. However, the results that have falsified the presumption have not completely falsified the hypothesis. There are two reasons why this is so. Firstly there were observed to be failings in the proxy ground truth that the results were tested against. Due to the method employed to create the proxy ground truth, each proxy ground truth included a very large number of rules. This meant it was biased against those results that produced fewer high quality rules by producing a very high number of false negatives – and so a low recall was obtained. This in turn meant that, if each of the higher fidelity models of classical conditioning restricted the number of rules, its recall would decrease regardless of how much the precision increased. As the recall results were much worse for every output, this biased the results of any measure that aimed to find a balance between the false positives and the false negatives. The initial presumption implied that an increase in the fidelity of the model would be due to an increase in precision with a more fixed recall. The precision results tentatively show that rise as the classical conditioning model fidelity increased.

The second and most important reason why the presumption failed but not necessarily the hypothesis is that the results provided tentative evidence in favour of the hypothesis. This evidence showed itself in a way other than the presumed correlation between fidelity of the model of classical conditioning and fidelity of the environmental model. This demonstrated by to two pieces of evidence. The first piece of evidence is that the three models not based upon classical conditioning, in general, did worse than the models that were based on classical conditioning. The second piece of evidence is that the general results for the system, regardless of which classical conditioning model was employed, produced some results that could be regarded as being in line with expectations. Many of the ideas of classical conditioning were used throughout the system – for example the associative behaviour conducted by the association module and the window system being based on the inter-stimulus interval. Because many of the ideas of classical conditioning are found throughout the system and not just in the models of classical conditioning, there can be seen to be evidence in favour of classical conditioning. As this evidence ignores the individual results of each model of classical conditioning, the presumption can fail without the hypothesis failing.

These two reasons as to why the presumption can fail but not the hypothesis do not lead to the conclusion that the hypothesis is deemed to be true however; while the results do include some evidence in favour of the hypothesis, this is not clear-cut enough to declare the hypothesis as standing nor to declare the hypothesis being falsified. Some of the suggestions for future work, discussed later on in this chapter, focus on improving the failings found within both the system and the manner in which the system was evaluated.

7.1.2 Hypothesis B

This hypothesis has held better than the first hypothesis. The evidence in favour of this hypothesis, as with some of the evidence in favour of the first hypothesis can be found in the fact that a system could be built in the first instance. The system itself passively observes an environment (in the case tested, various real-world physical effects) and learns a model of commonsense knowledge regarding the observations made.

While the system does show that a passive system is able to learn some information about its environment using conditioning, as with hypothesis A, the results have removed some of the clarity from the evidence in favour of this hypothesis. This clarity has been reduced in two ways. The first way is the result regarding the event type hierarchy. In building an event type hierarchy it was assumed that the event types would be organised as a collection of pyramids, where the top of the pyramid represented a maximal sequence of event types that cannot be added to due to the unpredictability of the connecting further event types. Each top-level event type would therefore be a complete frequently observed episode. An episode would be, in the case of the learning scenarios "throwing an object in the air", "one complete rotation of the objects" or "a collision between two balls". For a more intuitive case, if the system was observing someone doing household chores, the top-level event types would be episodes of regular sequences of activities that follow one another – for example "clean the carpet" and "a trip to the local shop". In addition a "clean the carpet followed by a trip to the local shop" high-level episode would not appear because of a low correlation between the two episodes occurring (unless the observed person has a very fixed routine for chores).

Because of the discovered incompatibility between an event type hierarchy and multi-frame event types leading to a cap on the number of levels in the hierarchy, the top level event types never form, except for extremely low peaks. This means that the fidelity of the learned model of the environment in general is severely curtailed as it is these top-level event types that completely represent the target concepts as an episode. The consequence for the hypothesis is that, while a partial model is still achieved with the level count cap in place, it is a much "more partial" model than was ever intended, and the learning scenarios were specifically chosen so that the system could potentially learn what could be regarded as a full model of the environment presented.

The second way where the results have removed some of the clarity in being able to state that the hypothesis stands are the results produced when comparing against the proxy ground truth. While the level one results were in line with expectations, they could have been better, and the overall results were less in line with expectations, though some of this can be attributed to the issues with the event type hierarchy. The failings in each of the proxy ground truth rule sets to completely and precisely depict their corresponding scenarios led to a potential reduction in how well the output of the system compared

with the proxy ground truth. As demonstrated in the results section, some of the rules output by the system that were not in the proxy ground truth could be interpreted as being identifiably true statements about their corresponding input scenario. This clouds the ability to state that the hypothesis stands because the results that were obtained indicate that in a wider sense, the system produces a commonsense model that is even "more partial" than was implied by the hypothesis and was discussed in chapter one. This is especially true considering that the scenarios were chosen to allow for a complete model to be potentially created by the system. The results being biased by an inaccurate proxy ground truth may have reduced the overall results for the system. Had the results that were obtained been better that they were, it could have been stated with more confidence that the hypothesis stands.

As was implied earlier, these two factors leading to a lower confidence in stating that the hypothesis stands are weaker than those same factors, weakening the conclusion for hypothesis A. The fact that the system works to any extent is confirmatory evidence of the hypothesis, however less than perfect results will always leave the possibility that the hypothesis could still be falsified.

7.2 Contributions

There are three main contributions of this thesis. These contributions are stated below and are then each discussed in turn.

1. The thesis has contributed a demonstration that the phenomena of classical conditioning could be used for an agent to learn a commonsense model of its environment.

2. The thesis has contributed a demonstration that a system using classical conditioning is able to passively learn about an observed environment.

3. The thesis has contributed an observation from analysis, which was subsequently demonstrated, that some of the phenomena of classical conditioning can be interpreted to exist in order to deal with the noise of happenstance that arises when learning associatively.

7.2.1 The First Contribution

In tentatively allowing hypothesis A to stand, for reasons discussed in the previous section, the thesis has demonstrated the possibility of using the phenomena of classical conditioning to learn a commonsense knowledge model of its environment. While this has only been shown in the more limited case of visual event types, there is no requirement for the event types to be visual – the only place within the system where it matters that the event types are of a visual nature is in the initial recognition of the atomic event types in the pre-processor module. If the pre-processor was changed for a different source

of atomic event types, there would be no changes needed within the remainder of the system. One of the pieces of future work looks at the possibility of using other sources and multiple sources of atomic event types.

The commonsense knowledge of the system is also very much incomplete compared to that knowledge expressed in human-made commonsense knowledge bases. For instance, a commonsense knowledge theory of gravity would also include, for example, the concept of objects supporting other objects. The knowledge learned by the system is much more limited in that example to "a ball moving up will move down". It is to be expected that the knowledge learned from a nascent system will be much more limited than can be achieved by a human explicitly encoding knowledge into a knowledge base. Human learning has access to a great many sources of event instances and information ranging from the base senses to higher sources of knowledge such as conversation and reading. This thesis represents a start on the path to developing a system capable of explicitly and directly learning all commonsense knowledge.

By learning a form of commonsense knowledge, rather than any form of value function, the thesis has contributed a distinctly different approach than that used by other reinforcement learning systems. While TD learning was created by making use of the same ideas and concepts, because that system learns a value function, the two systems are distinctly different.

7.2.2 The Second Contribution

The construction of the system was designed in such a manner that any knowledge learned was due to passive observation of the presented environment, rather than through interaction with the environment. By demonstrating that a system can be constructed that learns in a conditioning style, it meant that hypothesis B was allowed to stand. Such a system is a departure from other conditioning-based systems – reinforcement learning systems – which learn a value function for the actions performed.

It is not claimed that it is possible for all knowledge regarding an environment can be obtained through passive observation alone. As was discussed in chapter one, there are two occasions where action is required to allow for learning to occur. The first occasion is where observable environmental states are only reachable through action of the agent. This first occasion could be due to the environment state only changing as a consequence of agent action. Another possibility for this first occasion could be due to the fact that the agent is the only actor in the environment able to perform a particular action, and performance of that action is a prerequisite for the environment to enter a particular state or sub-set of states. The second occasion where action is required is where particular states of the environment require the agent to change its perspective to observe a particular aspect of an environmental state – for example the system will not learn from a person throwing a ball in the air if the camera is pointed in the wrong direction.

7.2.3 The Third Contribution

This thesis did not set out to make any commentary on the ideas, concepts and phenomena of classical conditioning itself, merely intending to make use of those ideas within a new artificial intelligence system. However, in the course of developing the thesis, it became apparent that a possible reason why some of the phenomena exist is to minimise the effect on learning that is due to the noise of happenstance.

The analysis of classical conditioning presented in chapter two of this thesis was written to explain why classical conditioning could be used to learn commonsense knowledge. From this analysis, it arose that, of the derived criteria for a passive learning system, the criterion for a passive system to have to deal with noise was by far the most common criterion for the phenomena reviewed to contribute towards. Of the twenty-eight phenomena reviewed, ten were argued to contribute towards minimising the effect of noise, with the mean number of phenomena contributing to a criterion being 3.6 and the second most commonly contributed towards criterion having six phenomena.

When the system was being built, several of the phenomena that were argued to contribute towards noise minimisation were included within both the models produced and some also inspired the design of the overall system. When the system was being tested, the noise robustness was specifically tested for. The results of the noise robustness test proved to be reasonably positive. This allowed for confirmation of the idea that from the perspective of the system, some of the phenomena of classical conditioning did contribute towards how robust the system was to the introduction of noise.

There were some results less in line with expectations for the noise robustness testing. Like the results that provided a level of doubt to the acceptance of the hypotheses, the results of the tests of the noise robustness less in line with expectations lend a level of doubt to the idea that the purpose of some of the phenomena of classical conditioning is to minimise noise. An example of a result that is less in line with expectations is the performance of the Inhibition and Pre-Exposure models within the system. Both models of classical conditioning introduced phenomena that were argued in the analysis to contribute to noise reduction. However, as discussed in the results chapter, even these results are reasonable overall.

In all the classical conditioning literature reviewed, not once was there any mention of the concept of noise or any other analogous concept. This does not necessarily mean that the concept is new within the context of classical conditioning; there are four possible interpretations for the absence of the concept. The first possibility is that the concept has appeared within parts of the literature that were not reviewed in the development of this thesis. However, given the relative importance of the concept within artificial intelligence, it would have been thought that some reference would have been made, at least in the literature surrounding models of classical conditioning.

The second possibility is that the contribution is obvious to researchers within the field of classical conditioning. This again is not a satisfactory explanation because if it were so obvious, then again it would have been thought that the concept or an analogous concept would have appeared at the very least as a side remark in the literature, given that the concept and its analogues can explain the animal behaviours observed.

The third possibility is that the concept was experimentally falsified as playing a role in classical conditioning – for example, by using subjects with brain lesions that don't exhibit a particular phenomenon still showing resilience to learning the noise). Other theories and concepts of classical conditioning have been falsified, such as the stimulus-substitution model advanced by Pavlov (1927) and the criticisms of the SOP model (Wagner & Brandon, 1989). But these issues were documented and discussed, including in textbooks of classical conditioning (Anderson, 2000; Klein, 1996). As there has been no reference, even to that of falsification, it again is not a satisfactory explanation.

The final possibility is that the concept of the noise of happenstance plays a significant part in determining the mechanism of classical conditioning. From the perspective of this thesis and the literature review of classical conditioning conducted, nothing has been found that contradicts this claim. However, given that this thesis never intended to make a contribution to the field of classical conditioning, there is doubt due to the unknown-unknowns that would be less valid in a similar contribution to artificial intelligence.

7.3 Future Work

Throughout the development of this thesis, a number of ideas for future work have become apparent. These ideas are presented below. Some of the ideas are direct and concrete plans for improvement of the system, some of the ideas are more nebulous and tangential. The order of the ideas presented roughly follows a progression from concrete to nebulous.

7.3.1 Revising the Hierarchical Event Type System

The issue that caused the hierarchical event type system to fail was a discovered incompatibility between event instances that occur over multiple frames and the use of hierarchical event types. However, the learning between individual pairs of atomic event types has provided reasonable results. There are currently two potential approaches that would allow for the build-up of more complex event types that would not suffer the same problem.

The first approach is that, instead of pairing event instances of equal level, the system should pair each atomic event instance with every confirmed composite event instance. This effectively means that the system learns whether an atomic event type should be appended on to a linear composite event type. The advantage is that only the maximal patterns would be stored. The chal-

lenge to the design would be that allowing re-use between composite event types would be difficult. Another challenge to be overcome would be the question of how to represent and learn the expected overlap-and-gap structure of the event types. This approach would be similar to that taken by Ivanov & Bobick (2000).

The second approach is to extend the event type system to use arbitrary sized groupings, rather than just pairs of event types. This means that where three or more event instances regularly mutually overlap, only a single rule would be created, stopping the infinite tower of event type levels that such a relationship creates within the current system. One possible approach to achieving a method of allowing the learning of arbitrary size groups is inspired by Grossberg & Schmajuk (1989) which can be interpreted to suggest that there should be separate processes to learn serial compound relationships and parallel compound relationships – this differentiation is described in section 7.3.2. Another possible approach, which could feed-into the first approach is to identify complete sub-graphs of an overlaps relation, where in the graph event instances are nodes and overlapping event type nodes have an edge. This second approach bears some similarity to ideas advanced by Sridhar, Cohn and Hogg (2008).

These two approaches could be augmented by another concept, which was also is inspired by Grossberg & Schmajuk (1989), who demonstrated in their model of classical conditioning an ability to learn not just predictions of an incoming unconditioned stimulus but also the timing of the stimulus. A similar approach of learning timings could be used in conjunction with either approach, where it would contribute to both of the previous approaches by allowing a method to represent and learn the overlap and gap structure of the composite event types.

7.3.2 Separate Processes for Parallel Compounds and Serial Compounds

Another possible augmentation of the system is again inspired by Grossberg & Schmajuk (1989). In the model presented in the aforementioned paper, it was demonstrated that the inter-stimulus interval curve can be constructed from a sigmoidal decay curve multiplied by a sigmoidal growth curve. This suggests that the inter-stimulus curve could be the product of competition between two separate processes. One possible competition could be between one process that attempts to "claim" event instances as being facets of the same event instance due to there being little or no time difference between the two and one process that attempts to "claim" event instances as being serial conjunctions where the first event instance is predictive of the second. With the inter-stimulus interval curve, there would be little need to learn to react to stimuli that are facets of the unconditioned stimulus, as they are not predictive; the curve is then a result of a compromise between the two processes.

Implementing this competition between processes would allow for the system to reduce the number of prediction rules created. Predictions could also become more accurate as when only one of two normally parallel compound event types is observed, the level of confidence in a prediction could be represented as being lower. This means that the impact of the resultant extinction could also be reduced, should the prediction be false. This system will have a bearing on how configural cues are implemented, as the process competing for "claiming" two event instances are facets of the same event instance could be the same system that decides how parallel compound stimuli are treated in terms of prediction.

7.3.3 The System as a Classifier

The focus of this thesis was to create a system that passively learned a model of its observed environment. This focus meant that the evaluation of the system looked at how accurate the environmental model it created matched that of a human conception of that environment. A different perspective is that this output of the different environmental models the system creates will contain event patterns that are unique to each scenario the system is presented with. Therefore, by comparing the list of environmental models of known scenarios, the system may be able to classify unknown scenarios.

By using the environmental models as essentially feature vectors, the system may be able to be used as a supervised machine learning system. This would allow the system to be evaluated in a more conventional manner and so be directly compared against existing machine learning systems.

The system in its current state would not be able to do this, as there would need to be an extension that creates a database of environmental models and a comparison system that allows comparison between two given environmental models, with some similarity metric. Given the potential size of each environmental model, it may be non-trivial to find an efficient system to do this. One potential way to extend the system to create easily comparable environmental models is to restrict each environmental model to only those rules that are unique to each classification, which would limit their size and so speed-up classification.

7.3.4 Generalisation and Discrimination

The implementation of these two phenomena is discussed separately from the discussion of implementing the other phenomena of classical conditioning due to their higher importance. When deciding which phenomena would be implemented, these two phenomena were the borderline cases. They were eventually not included due to having limited time and the amount of implementation time the two phenomena would need, meant that they had to be left out.

The reason for the amount of time needed is that these two phenomena require that learned objects are no longer atomic symbols, but instead will have

to be comprised of a composite of other lower-level data. In generalisation and discrimination, there needs to be a measure of how similar any two objects are so that a decision to whether to generalise an association to another object can be made.

There is a nascent idea for a possible generalisation and discrimination system that would fit with the system described in the thesis. For these systems, event types are conditioned as before, but patterns of event types are allowed to apply to more than one type of object, based on the generalisation and discrimination system selected. The nascent generalisation system is in part based on a system proposed by Alonso, Mondragón & Kjäll-Ohlsson (2006). Each object comprises of a feature vector. When two event types are paired together, a Gaussian curve around each value of the feature vector is grown. When a prediction of a compound event type occurring is being made, predictions are based on the event type and the similarity of the object to the feature vector of each rule, based on the Gaussian curves. Discrimination can be implemented in this manner by changing the Gaussian curve to that of a Gaussian Mixture Model – a technique that has shown success within computer vision (Stauffer & Grimson, 1999). In the Gaussian Mixture Model, when a negative prediction occurs, a negative Gaussian curve is grown around the values of the feature vector. This means that in the overall curve for the Gaussian Mixture Model, any exceptions are less likely to match the prediction pattern.

Another more nebulous idea for implementing generalisation and discrimination while still retaining object symbols is for the system to independently learn an ontology of observable objects. When the same compound event type is observed to occur using different objects, the objects within the compound event type recognition rule could then be replaced with the common ancestor within the learned ontology. Discrimination could be implemented as a list of exceptions to the learned rule; again this exception list could be expanded on the basis of "the least general generaliser" as with the main object of the recognition rule. A possible route for creating a system to learn an ontology would be to follow the functional object category learning described by Sridhar, Cohn & Hogg (2008).

7.3.5 Further Phenomena of Classical Conditioning

As more phenomena of classical conditioning are considered, both for inclusion within the model system, or through other mechanisms added to the wider system, the design of the system is forced to take into account wider forms of knowledge that could be learned, and the accuracy with which that knowledge is learned. Inclusion of further phenomena cannot be taken to guarantee improvement of the results. However, by considering many of the remaining phenomena of classical conditioning in turn, in a machine learning context, this can lead to ideas on how the implementations of phenomena that have previously been implemented could improve.

While generalisation, discrimination and to a lesser extent configural cues have been given some consideration on how they might by implemented, this is not yet true of the other phenomena. Of the remaining phenomena, partial reinforcement and the partial reinforcement extinction effect would be the next most important to consider, as implementing those could allow the system to be expanded to deal with more stochastic event type sequences.

7.3.6 Learnable Pre-Processor Atomic Events

The system as presented will always be limited in scope by the atomic event types that have been defined. For a more general learning system, these atomic event types will eventually need to be learned from the input data. While there will always be a need for primitives of some description, the more basic these are made, the more general the learning can be.

In appendix A, the atomic event type definitions listed are all based on changes in the state of the environment between consecutive frames. Instead of explicitly declaring the separate ways in which a state can change, these transitions could be learnable. One possible method would be to implement a similar conditioning method as the system that builds-up sequences in a spatial domain rather than a temporal domain. A second possible method would be to simply use the states as the atomic event types, though this method would require the system in general to learn ordering constraints on the patterns learned, so that directions can be distinguished where needed.

7.3.7 A Fully Real-Time Version of the System

Currently, the system is only partially real-time, in that the input data is real-time but the system stores event instances in memory until they are complete, meaning that the association module and the significance module operate in a trial-level manner. A fully real-time system would operate throughout in a real-time manner. The advantage in doing this is two-fold. The first advantage is that the computational performance of the system would likely be improved, especially if it could be implemented such that some operations can occur in parallel. The second advantage would be that the system would increase in biological plausibility, which means that the system could potentially begin to give more insight into biological conditioning.

One possible method that could be a part of this effort would be to replace the moving window with some form of activation trace applied to each rule. This would be similar to the eligibility trace used throughout the field of reinforcement learning. Another possible method that should be considered is that instead of having event instances that occur over intervals, every event instance is composed of two instances: a single-frame onset event instance and a single-frame offset event instance. Basing prediction on onsets and offsets was first suggested by Mowrer (1960) and is used within the temporal difference model of classical conditioning (Sutton & Barto, 1990).

There is another possible extension of the system related to moving the system to be fully-real time: The integration of a real-time tracker directly in to the system. This would allow for any measure of uncertainty that is produced by the tracking system to be used, which, when combined with the predictions of the recognition system, could allow for the accuracy of the track to be improved.

7.3.8 A Return to Actions and Rewards

The system presented by this thesis can be seen as a system attempting to predict the consequent event instances given the set of current event instances. This could easily be extended to include atomic event types that represent action event types and reward event types. The system would learn the consequences of action event types as if they were any other event type. With more complex composite event types, reward atomic event types could be included in the consequent event types of an action atomic event type. This would mean that every learned composite event type could be given a sum for its total reward value, based on the reward atomic event types that the composite event type contains. This would mean that the composite event types that include an action atomic event type and have a reward atomic event type could be used for action selection, by selecting an action with a maximal reward. This would allow the system to also learn to predict the actions it will select in the future, given the observed external event instances.

In a traditional reinforcement learning system, the state of the environment gives rise to the actions being selected. This system would not make use of states in that manner for action selection. Instead, it can be seen that the system would observe the environment until it could predict a plan of action that would lead it to a reward. This plan of action would be followed, but as the narrative of the atomic event instance stream unfolded, the current plan of action would be corrected based upon new predictions. Should the stream of event instances indicate a larger reward than the current plan, the plan would be adapted. This system would also allow for the need to plan timings of actions so that they have the desired effect based upon the how the dynamic environment changes. While it would require theoretical work to confirm, it is conjectured that this style of action selection would mean that the Markov property would not need to be assumed.

7.3.9 Wider Testing of the System

The system has currently been tested on only three learning scenarios using one set of atomic event types. There are many ways in which the system can be tested, both using the same set of atomic event types and using a different set of atomic event types. The first task would be to repeat several times the processing for differently generated videos for the same set of experiments, video durations, noise levels and models as the results presented in this thesis.

By doing this, and taking the mean of the results, the results would be given a more robust standing.

A second simple set of additional testing would be to continue to use tracks of real-world video footage of the existing learning scenarios. By comparing the results of real-world tracker and the simulated tracker at different noise levels, a characterisation of the noise levels that are present in the real-world tracker could be given.

The third source of extra testing could be found in attempting to adapt the system in general to the three learning scenarios presented in chapter five that were not used: Fragile objects shattering, two people passing a ball and rigid object configuration spaces. Some of the other pieces of further work presented in this section would help with this task, though there may be work required on top of what is discussed in this section to allow the system to learn in all six scenarios – particularly in the work needed to learn a configuration space.

A fourth source of testing is inspired by Pavlov's (1927) experiments with dogs. In the experiments, Pavlov measured the amount of saliva by surgically attaching a collection tube near the dog's salivary gland. It could be possible to replicate these experiments on computer. This could be done by making two relatively simple modifications to the system. The first would be the creation of a privileged single-frame atomic event that is directly encoded in the input, rather than detected by the pre-processor. This privileged atomic event would represent an unconditioned stimulus. The second modification would be to add a counter variable. When the privileged atomic event is encountered or predicted, the counter is increased by the strength of the prediction (with the privileged atomic event having a fixed value). This counter variable would represent the amount of saliva accumulated in the collection device – i.e. the conditioned and unconditioned responses. In this manner, the system can be tested as if it were the subject of a classical conditioning experiment. By experimenting with the system in this manner, the models can be tested as part of the system, rather than just the simple bug-testing that was done in the development of this thesis, to make sure the significance model produced the expected response.

Another source would be to extend the system to add auditory atomic event types. Initially the auditory event types could be specific spoken words, though later if testing was successful enough, the word event types could be replaced with phoneme event types. These additions could allow the system to learn rules for simple games such as snap. An existing system that learns rules of simple games is described by Needham *et al.* (2005), which learns using an inductive logic programming system known as Progol (Muggleton, 1995). Progol is a supervised, offline learning system, and so replacing the learning element of the system by Needham *et al.* (2005) would allow the system to learn the rules as the game was being observed.

The final proposed test is to apply the system to a typical application area that makes use of event sequence learning. This is the area of automated surveillance, which uses event sequence learning to predict if the behaviour of a tracked individual is going to lead to a crime. One example of this sort of behaviour is someone approaching the door of multiple cars in a car park. The system proposed in this thesis, once improved, should be able to learn sequences of event types that include a "crime" privileged atomic event type, which acts as a label for any preceding pattern learned. When the system predicts a crime event instance is imminent, then the relevant person could be alerted for the prediction to be checked and for any required action to be taken.

7.3.10 A New Proxy Ground Truth Method

In the analysis of the results, it was found that the proxy ground truth used was flawed. The proxy ground truth had too many rules, biasing the results towards indiscriminate significance models, and the proxy ground truth missed out rules that could be argued to be legitimate. The blame for these issues was placed on the method used to generate the proxy ground truth rule sets. The method used seemed sound at the time as it combined steps that used human intelligence with deterministic steps in an attempt to avoid the problems of introspection. The concept of using a mix of deterministic and human steps is still believed to be the best approach to the creation of a proxy ground truth. However, the actual steps that are used need to be changed. It is unknown what should be done differently. A thorough review of the steps and how this led to a very large set of rules would be the first action taken in creating a new method for creating a proxy ground truth.

There is a wider issue that is highlighted by this example of an inaccurate proxy ground truth. As artificial intelligence systems become ever more complex, any proxy ground truths that attempt to evaluate the internal knowledge of those systems will grow in complexity too. This means that at some point of complexity, the feasibility of accurately and completely evaluating the abilities and knowledge of artificial intelligence systems will become ever more difficult. Looking at this proxy ground truth complexity issue and finding solutions to it may be a worthwhile topic of research.

7.3.11 Stochastic Outcomes

Currently, the system produces rules where there is a single possible predicted outcome to a rule. It would be more general if a system could learn rules where the outcome is one of a set of possibilities, with an associated probability distribution. Consider the act of rolling a die, there is a set number of event instances that could follow, but that list is finite. The difficulty is how to discriminate between an outcome event instance that should be considered a member of the possible outcome set and one that should be rejected as noise.

The learning of these stochastic outcomes will most likely be related to any developments of the partial reinforcement and partial reinforcement extinction effect phenomena. There is a difference between the idea of multiple stochastic outcomes and the aforementioned phenomena – partial reinforcement refers to there being only a single prediction that may or may not be true, rather than a wider set of possibilities. It would be of interest for there to be animal experimentation where a single conditioned stimulus is probabilistically paired with several different unconditioned stimuli that are known to elicit different and preferably mutually exclusive conditioned responses. Would the subject alternate between responses, attempting to second-guess the stochastic process or would there be a dominant response? How does the animal accommodate entirely unrelated unconditioned stimuli? No literature describing such experiments has been found.

7.3.12 Subjective Probability

As was described in chapter two, some of the phenomena when combined together allow for the contingency phenomenon as an epiphenomenon. Due to this, the association strength can be seen to be analogous to a conditional probability. However, because other factors such as timing and magnitude are involved, this is not a pure conditional probability, but can be seen as being more "subjective". This leads to the concept that the irrational probabilistic reasoning seen in humans, in such phenomena as those believing in winning streaks, could in fact be a computational trade-off.

The subjective concept of randomness could be based on an expectation of the law of large numbers eventually coming into effect. Where a long enough run of the same outcome is observed, or another outcome is not observed for a long time, there could be a loss in the belief that the outcome is random. This suggests a mechanism by which noise and one or more stochastic outcomes can be distinguished, as by the law of large numbers, genuine members of the stochastic outcome set would be observed again whereas this is not the case for instances of noise. With an objective conditional probability, any outcome, noise or otherwise, could not be removed from the list of possible outcomes, as each outcome has already been observed. The implication for the irrational behaviour evidenced in the winning streak is that if the mind was not susceptible to the concept of a winning streak, it would not be able to learn stochastic patterns in the first place.

7.3.13 A Multiple-Environment Reinforcement Learning Problem

During the review of the relevant literature of reinforcement learning, an idea was conceived for possible future research. Consider the scenario where the agent is not a singular organism, but instead is an organisation. This organisation could be operating in multiple countries simultaneously. Each country

shares some similarities but also could have some differences. In this scenario, if the agent learned in a reinforcement leaning style, it is able to select multiple actions simultaneously. In this case, one could take the set of states for each environment and combine them to create a singular "global" set of environmental states, with a similar set for the sets of actions. This set-up though would not easily be able to exploit the similarities of each environment. If instead, the differences of each environment are modelled, then the system would be able to use the similarities and a model of the differences to be able to explore one environment while maximising the currently best-known policy in other environments. If the environment that is being used for exploration produces a better policy that, when the differences between environments are taken into account, could work in the other environments, then the new policy could be attempted. This would allow for another type of balance between exploration and exploitation of policy. The research for this would likely have to look at: How best to learn the differences between each environment, policies to select which environment to explore and meta-policies for rolling-out potential new policies into new environments.

7.3.14 Ideas Concerning Classical Conditioning

During the development of this thesis, a few ideas were conceived that could lead to fruitful research within the field of classical conditioning. The first of these ideas was conceived during the analysis of the contingency phenomenon. During that analysis, the following derivation was found:

$$P\left(B|A\right) - P\left(B|\overline{A}\right) = \frac{P\left(A \cap B\right)}{P\left(A\right)} - \frac{P\left(\overline{A} \cap B\right)}{P\left(\overline{A}\right)} \tag{7.1}$$

$$= \frac{P\left(A \cap B\right)}{P\left(A\right)} - \frac{P\left(B\right) - P\left(A \cap B\right)}{1 - P\left(A\right)} \tag{7.2}$$

$$= \frac{\frac{|A \cap B|}{|\Omega|}}{\frac{|A|}{|\Omega|}} - \frac{\frac{|B|}{|\Omega|} - \frac{|A \cap B|}{|\Omega|}}{1 - \frac{|A|}{|\Omega|}} \tag{7.3}$$

$$= \frac{|A \cap B|}{|A|} - \frac{|B| - |A \cap B|}{|\Omega| - |A|} \tag{7.4}$$

In this derivation, Ω denotes the set of all observed event instances. The implication of the final equation of the derivation is that it is likely that the age of the subject determines how conditioning functions. This is because of the presence of the $|\Omega|$ in the second term of the equation, which can be interpreted as being the count of all observed stimuli that the subject has ever experienced, regardless of whether event types A or B are present. This implies that as the subject ages, the total amount of experience the subject has increases, and so the second term of the equation becomes increasingly fixed as it becomes dominated by the total experience. This implies that extinction of a conditioned response based on singular presentations of the unconditioned stimulus could be possible if the subject is young enough. It is

usually not possible to extinguish an association through presentation of the unconditioned stimulus, so if it is possible in young subjects, insight into how the contingency phenomenon works would be gained.

The second idea concerns a thought experiment. Consider a wind-up toy or a catapult. The longer the "wind-up" event instance is, one expects that the magnitude of the corresponding "release" event instance will be greater. This basic idea leads to the question: is this an expectation that can be learned through conditioning? If the subject is presented with an environment where the length of the inter-stimulus interval determined the magnitude of the unconditioned stimulus, does the compensatory nature of the conditioned response increase in intensity for longer inter-stimulus intervals?

7.3.15 Ideas Concerning Bayesian Learning

This thesis has not made use of the multitude of Bayesian artificial intelligence techniques, even though using Bayes' rule for the significance measure would be a natural choice. This was deliberate in order avoid allowing the argument that the system is a new learning system to be subjected to the counterargument that because the system makes use of Bayes, an existing form of learning systems, it is not a new learning system in its own right. Now that the system has been established to be able to learn without the need for a Bayesian significance measure, it would be a natural extension to include a Bayesian significance measure into the system.

Due to the knowledge that a Bayesian significance measure would be a natural extension of the system, during the development of this thesis, a few of insights into Bayesian learning were conceived. The first idea asks if it would be possible, using a similar technique used to create an iterative sigmoid function, to create an iterative Bayes' rule. For an iterative Bayes' rule there would have to be two functions, one for reinforcement and one for non-reinforcement.

The second insight concerns the observation that the association strength can be seen to be the time-discounted risk of an event type occurring. The measure is time-discounted due to the influence of the inter-stimulus interval and is risk rather than probability because it is influenced by magnitude. In learning patterns of event types, Bayes' rule can be used to predict the probability that an event instance will precede another. It could be potentially rewarding to include measures of both the timing and magnitude into a variant of Bayes' rule.

7.3.16 A Universal-Mode Learning System

Machine learning has traditionally been presented as being composed of three modes of input data. The first is supervised learning, where the agent is given a piece of data and is expected to apply the labelled data provided to predict the label for unseen data. The second mode is unsupervised learning, where the agent is given data with no label and is expected to create its own labels

based on a derived structure of the data. The third mode is reinforcement learning, where the agent predicts the label (an action) for some input data (a state) and then is told whether the produced label is correct or incorrect, possibly after a batch of action-state pairings.

There exists systems that are in-between these three systems, such as semi-supervised learning (Xiaojin, 2008; Grira *et al.*, 2004), which lies between supervised and unsupervised learning. Other systems, such as gradient-descent temporal difference learning (Sutton & Barto 1998, pp. 193–226; Sutton *et al.* 2009) integrate supervised systems into reinforcement learning, in order for reinforcement learning to generalise its experience to new states. The system presented in this thesis can be argued to lie between unsupervised and reinforcement learning.

It is proposed that it would be possible to build a system that is able to learn in each of the three modes of learning. The proposed system would be able to combine data from each of the three modes of learning. The basis of such a system could be a far descendent of the system proposed in this thesis. With such a basis, each of the modes of learning could be used. Traditional reinforcement learning could be dealt with through action and reward event types, as described previously. Supervised learning would be a case of learning to associate between the input data and the label; the source of generalisation by the system is previously described. Unsupervised learning would be able to be achieved through the build-up of patterns presented together, both spatially and temporally. As humans are capable of learning in each mode, if the eventual goal of artificial intelligence is the creation of a human-level intelligent agent, then work to merge the three modes will eventually need to happen. The brief ideas presented here may be a very small start in that direction.

7.3.17 Ultra-High Fidelity Environmental Modelling

The final idea of this thesis will be very complex, and require a great deal of extra research. The idea is that it may be possible to create video footage of future predictions based upon an ultra-high fidelity learned model of the environment. The mind acts to be able to predict what the low-level sensations would be for an action. In the imagination, the consequent shapes, sounds and colours of an action that have not yet been performed can be predicted. Yet these imaginary images and sounds have a level of fidelity below that of when the same images and sounds are presented in reality, making them feel elusive compared to reality.

A complete learning system learns how to recognise objects and their movement from pixels of a video and is able to learn repeated patterns of movements and object interactions. This means it has a model of those very same objects and their movements. Therefore it should be possible to reverse the learned model to produce a rendered video of a particular movement. When a system learns patterns of movements, those patterns would similarly be able to create a rendered video.

The primary feedback loop of the system described by this thesis is that low-level event instances construct high-level event types and then the high-level event types aid prediction of any low-level event instances. If the agent is able to learn a model of the environment that is of high enough fidelity – as discussed in section 7.3.6, it would also be able be augmented to use this feedback loop over many hierarchical levels to be able to predict specific pixel values from high-level abstract descriptions such as "someone throwing a ball into the air", filling in the relevant detail to the current situation as the hierarchy descends to create the individual pixel predictions. Similarly, a system able to learn a pixel-level model of the environment would be able to take the stream of pixel data and construct ever higher levels of event type sequences, abstracting away from detail as the hierarchy ascends.

Such a system would allow for much simpler testing of how the system has learned. Instead of having to create a proxy ground truth, the input data itself is the ground truth. When testing a learning system, all that would be needed would be to begin presenting a video to the hypothetical system and part way through stop showing the video and compare the pixel-level prediction of the system with the actual video. The longer it maintains an accurate description of the video, the better the system has become at learning.

There is another work that discusses feedback loops occurring over hierarchies. Hofstadter (1979; 2007) argued that the essential nature of consciousness was a loop that occurs over a hierarchy; a loop Hofstadter called a strange loop. This strange loop was such that when the hierarchy is followed in either an upwards or downwards manner, the path always returns to the beginning. When attempting to predict the future at a given level of abstraction, this entails the predication of some of the details at the lower level of abstraction, which entails lower still levels of prediction. However, the purpose of creating predictions of high-level event instances is ultimately to better predict and plan to achieve the highest-level of the hierarchy: the overall goal event types of the system. Those same goals however are the low-level reward and punishment event types. Therefore, the highest-level goal event types are the same as the lowest-level reward and punishment event types and vice-versa. It is therefore argued that a planning and prediction feedback loop over the hierarchy of an ultra-high environmental model may be a candidate to be called a strange loop.

Appendix A

Atomic Event Definitions

These are the definitions of the atomic events in terms of consecutive frame-state variables. Atomic events are used as the primitive symbols that the system learns its event patterns from. An explanation of the notation can be found within chapter four, most notably section 3.2.1.1 and section 4.2.2. A discussion of the generation of these atomic events is in section 4.2.3. The definitions that are directly based on the event definitions by dos Santos *et al.* (2009) are starred on the left side of the page. It is noted that these influenced the other events that were chosen.

In order to reduce the amount of repetition within these definitions, the here predicate is defined to be true when all members of the input list are visible at both times:

$$\begin{aligned}&\texttt{here}\left(ObjectList, TimeList\right) \longleftrightarrow \\ &\quad \forall o \in ObjectList, \forall t \in TimeList\ \texttt{holdsAt}\left(\texttt{visible}\left(o\right), t\right)\end{aligned} \tag{A.1}$$

The here predicate is not actually used within the system and is only defined to allow for the atomic event definitions listed here to be shorter. The remainder of this appendix lists the atomic event definitions.

$$\begin{aligned}&\texttt{happens}\left(\texttt{lost}\left(o\right), [t_1, t_2]\right) \longleftrightarrow \\ &\qquad \texttt{holdsAt}\left(\texttt{visible}\left(o\right), t_1\right) \wedge \\ &\qquad \neg\texttt{holdsAt}\left(\texttt{visible}\left(o\right), t_2\right)\end{aligned} \tag{A.2}$$

$$\texttt{happens}\left(\texttt{found}\left(o\right),\left[t_1,t_2\right]\right)\longleftrightarrow$$
$$\neg\texttt{holdsAt}\left(\texttt{visible}\left(o\right),t_1\right)\wedge$$
$$\texttt{holdsAt}\left(\texttt{visible}\left(o\right),t_2\right) \quad \text{(A.3)}$$

$$\texttt{happens}\left(\texttt{moveLeft}\left(o\right),\left[t_1,t_2\right]\right)\longleftrightarrow$$
$$\texttt{here}\left(\left[o\right],\left[t_1,t_2\right]\right)\wedge$$
$$\texttt{holdsAt}\left(\text{pos}_x\left(o,x_1\right),t_1\right)\wedge$$
$$\texttt{holdsAt}\left(\text{pos}_x\left(o,x_2\right),t_2\right)\wedge$$
$$x_1 > x_2 \quad \text{(A.4)}$$

$$\texttt{happens}\left(\texttt{moveRight}\left(o\right),\left[t_1,t_2\right]\right)\longleftrightarrow$$
$$\texttt{here}\left(\left[o\right],\left[t_1,t_2\right]\right)\wedge$$
$$\texttt{holdsAt}\left(\text{pos}_x\left(o,x_1\right),t_1\right)\wedge$$
$$\texttt{holdsAt}\left(\text{pos}_x\left(o,x_2\right),t_2\right)\wedge$$
$$x_1 < x_2 \quad \text{(A.5)}$$

$$\texttt{happens}\left(\texttt{moveUp}\left(o\right),\left[t_1,t_2\right]\right)\longleftrightarrow$$
$$\texttt{here}\left(\left[o\right],\left[t_1,t_2\right]\right)\wedge$$
$$\texttt{holdsAt}\left(\text{pos}_y\left(o,y_1\right),t_1\right)\wedge$$
$$\texttt{holdsAt}\left(\text{pos}_y\left(o,y_2\right),t_2\right)\wedge$$
$$y_1 > y_2 \quad \text{(A.6)}$$

$$\texttt{happens}\left(\texttt{moveDown}\left(o\right),\left[t_1,t_2\right]\right)\longleftrightarrow$$
$$\texttt{here}\left(\left[o\right],\left[t_1,t_2\right]\right)\wedge$$
$$\texttt{holdsAt}\left(\text{pos}_y\left(o,y_1\right),t_1\right)\wedge$$
$$\texttt{holdsAt}\left(\text{pos}_y\left(o,y_2\right),t_2\right)\wedge$$
$$y_1 < y_2 \quad \text{(A.7)}$$

$$\star \quad \texttt{happens}\left(\texttt{approaching}\left(o_1, o_2\right), \left[t_1, t_2\right]\right) \longleftrightarrow$$

$$\begin{aligned} & \texttt{here}\left(\left[o_1, o_2\right], \left[t_1, t_2\right]\right) \wedge \\ & \left(\begin{array}{l} \texttt{holdsAt}\left(\texttt{disC}\left(o_1, o_2\right), t_1\right) \vee \\ \texttt{holdsAt}\left(\texttt{extC}\left(o_1, o_2\right), t_1\right) \end{array} \right) \wedge \\ & \neg\texttt{holdsAt}\left(\texttt{co}\left(o_1, o_2\right), t_2\right) \wedge \\ & \texttt{holdsAt}\left(\texttt{dist}\left(o_1, o_2, d_1\right), t_1\right) \wedge \\ & \texttt{holdsAt}\left(\texttt{dist}\left(o_1, o_2, d_2\right), t_2\right) \wedge \\ & d_1 > d_2 \end{aligned} \quad \text{(A.8)}$$

$$\star \quad \texttt{happens}\left(\texttt{receding}\left(o_1, o_2\right), \left[t_1, t_2\right]\right) \longleftrightarrow$$

$$\begin{aligned} & \texttt{here}\left(\left[o_1, o_2\right], \left[t_1, t_2\right]\right) \wedge \\ & \left(\begin{array}{l} \texttt{holdsAt}\left(\texttt{disC}\left(o_1, o_2\right), t_1\right) \vee \\ \texttt{holdsAt}\left(\texttt{extC}\left(o_1, o_2\right), t_1\right) \end{array} \right) \wedge \\ & \texttt{holdsAt}\left(\texttt{dist}\left(o_1, o_2, d_1\right), t_1\right) \wedge \\ & \texttt{holdsAt}\left(\texttt{dist}\left(o_1, o_2, d_2\right), t_2\right) \wedge \\ & d_1 < d_2 \end{aligned} \quad \text{(A.9)}$$

$$\star \quad \texttt{happens}\left(\texttt{mergeRight}\left(o_1, o_2\right), \left[t_1, t_2\right]\right) \longleftrightarrow$$

$$\begin{aligned} & \texttt{here}\left(\left[o_1, o_2\right], \left[t_1, t_2\right]\right) \wedge \\ & \left(\begin{array}{l} \texttt{holdsAt}\left(\texttt{disC}\left(o_1, o_2\right), t_1\right) \vee \\ \texttt{holdsAt}\left(\texttt{extC}\left(o_1, o_2\right), t_1\right) \end{array} \right) \wedge \\ & \texttt{holdsAt}\left(\texttt{left}\left(o_1, o_2\right), t_1\right) \wedge \\ & \texttt{holdsAt}\left(\texttt{co}\left(o_1, o_2\right), t_2\right) \end{aligned} \quad \text{(A.10)}$$

$$\star \quad \texttt{happens}\left(\texttt{mergeLeft}\left(o_1, o_2\right), \left[t_1, t_2\right]\right) \longleftrightarrow$$

$$\begin{aligned} & \texttt{here}\left(\left[o_1, o_2\right], \left[t_1, t_2\right]\right) \wedge \\ & \left(\begin{array}{l} \texttt{holdsAt}\left(\texttt{disC}\left(o_1, o_2\right), t_1\right) \vee \\ \texttt{holdsAt}\left(\texttt{extC}\left(o_1, o_2\right), t_1\right) \end{array} \right) \wedge \\ & \texttt{holdsAt}\left(\texttt{left}\left(o_2, o_1\right), t_1\right) \wedge \\ & \texttt{holdsAt}\left(\texttt{co}\left(o_1, o_2\right), t_2\right) \end{aligned} \quad \text{(A.11)}$$

$$\texttt{happens}\left(\texttt{mergeTop}\left(o_1, o_2\right), \left[t_1, t_2\right]\right) \longleftrightarrow$$
$$\texttt{here}\left(\left[o_1, o_2\right], \left[t_1, t_2\right]\right) \wedge$$
$$\left(\begin{array}{l}\texttt{holdsAt}\left(\texttt{disC}\left(o_1, o_2\right), t_1\right) \vee \\ \texttt{holdsAt}\left(\texttt{extC}\left(o_1, o_2\right), t_1\right)\end{array}\right) \wedge$$
$$\texttt{holdsAt}\left(\texttt{above}\left(o_1, o_2\right), t_1\right) \wedge$$
$$\texttt{holdsAt}\left(\texttt{co}\left(o_1, o_2\right), t_2\right) \quad \text{(A.12)}$$

$$\texttt{happens}\left(\texttt{mergeBottom}\left(o_1, o_2\right), \left[t_1, t_2\right]\right) \longleftrightarrow$$
$$\texttt{here}\left(\left[o_1, o_2\right], \left[t_1, t_2\right]\right) \wedge$$
$$\left(\begin{array}{l}\texttt{holdsAt}\left(\texttt{disC}\left(o_1, o_2\right), t_1\right) \vee \\ \texttt{holdsAt}\left(\texttt{extC}\left(o_1, o_2\right), t_1\right)\end{array}\right) \wedge$$
$$\texttt{holdsAt}\left(\texttt{above}\left(o_2, o_1\right), t_1\right) \wedge$$
$$\texttt{holdsAt}\left(\texttt{co}\left(o_1, o_2\right), t_2\right) \quad \text{(A.13)}$$

$\star$ $\texttt{happens}\left(\texttt{emergeRight}\left(o_1, o_2\right), \left[t_1, t_2\right]\right) \longleftrightarrow$
$$\texttt{here}\left(\left[o_1, o_2\right], \left[t_1, t_2\right]\right) \wedge$$
$$\texttt{holdsAt}\left(\texttt{co}\left(o_1, o_2\right), t_1\right) \wedge$$
$$\left(\begin{array}{l}\texttt{holdsAt}\left(\texttt{disC}\left(o_1, o_2\right), t_2\right) \vee \\ \texttt{holdsAt}\left(\texttt{extC}\left(o_1, o_2\right), t_2\right)\end{array}\right) \wedge$$
$$\texttt{holdsAt}\left(\texttt{left}\left(o_1, o_2\right), t_2\right) \quad \text{(A.14)}$$

$\star$ $\texttt{happens}\left(\texttt{emergeLeft}\left(o_1, o_2\right), \left[t_1, t_2\right]\right) \longleftrightarrow$
$$\texttt{here}\left(\left[o_1, o_2\right], \left[t_1, t_2\right]\right) \wedge$$
$$\texttt{holdsAt}\left(\texttt{co}\left(o_1, o_2\right), t_1\right) \wedge$$
$$\left(\begin{array}{l}\texttt{holdsAt}\left(\texttt{disC}\left(o_1, o_2\right), t_2\right) \vee \\ \texttt{holdsAt}\left(\texttt{extC}\left(o_1, o_2\right), t_2\right)\end{array}\right) \wedge$$
$$\texttt{holdsAt}\left(\texttt{left}\left(o_2, o_1\right), t_2\right) \quad \text{(A.15)}$$

$$\texttt{happens}\left(\texttt{emergeTop}\left(o_1, o_2\right), \left[t_1, t_2\right]\right) \longleftrightarrow$$
$$\texttt{here}\left(\left[o_1, o_2\right], \left[t_1, t_2\right]\right) \wedge$$
$$\texttt{holdsAt}\left(\texttt{co}\left(o_1, o_2\right), t_1\right) \wedge$$
$$\left(\begin{array}{l}\texttt{holdsAt}\left(\texttt{disC}\left(o_1, o_2\right), t_2\right) \vee \\ \texttt{holdsAt}\left(\texttt{extC}\left(o_1, o_2\right), t_2\right)\end{array}\right) \wedge$$
$$\texttt{holdsAt}\left(\texttt{above}\left(o_1, o_2\right), t_2\right) \quad \text{(A.16)}$$

$$\texttt{happens}\left(\texttt{emergeBottom}\left(o_1,o_2\right),\left[t_1,t_2\right]\right)\longleftrightarrow$$
$$\begin{aligned}&\texttt{here}\left(\left[o_1,o_2\right],\left[t_1,t_2\right]\right)\wedge\\&\texttt{holdsAt}\left(\texttt{co}\left(o_1,o_2\right),t_1\right)\wedge\\&\left(\begin{array}{l}\texttt{holdsAt}\left(\texttt{disC}\left(o_1,o_2\right),t_2\right)\vee\\\texttt{holdsAt}\left(\texttt{extC}\left(o_1,o_2\right),t_2\right)\end{array}\right)\wedge\\&\texttt{holdsAt}\left(\texttt{above}\left(o_2,o_1\right),t_2\right)\end{aligned}\tag{A.17}$$

$$\texttt{happens}\left(\texttt{makeContactRight}\left(o_1,o_2\right),\left[t_1,t_2\right]\right)\longleftrightarrow$$
$$\begin{aligned}&\texttt{here}\left(\left[o_1,o_2\right],\left[t_1,t_2\right]\right)\wedge\\&\texttt{holdsAt}\left(\texttt{disC}\left(o_1,o_2\right),t_1\right)\wedge\\&\texttt{holdsAt}\left(\texttt{left}\left(o_1,o_2\right),t_1\right)\wedge\\&\left(\begin{array}{l}\texttt{holdsAt}\left(\texttt{extC}\left(o_1,o_2\right),t_2\right)\vee\\\texttt{holdsAt}\left(\texttt{co}\left(o_1,o_2\right),t_2\right)\end{array}\right)\end{aligned}\tag{A.18}$$

$$\texttt{happens}\left(\texttt{makeContactLeft}\left(o_1,o_2\right),\left[t_1,t_2\right]\right)\longleftrightarrow$$
$$\begin{aligned}&\texttt{here}\left(\left[o_1,o_2\right],\left[t_1,t_2\right]\right)\wedge\\&\texttt{holdsAt}\left(\texttt{disC}\left(o_1,o_2\right),t_1\right)\wedge\\&\texttt{holdsAt}\left(\texttt{left}\left(o_2,o_1\right),t_1\right)\wedge\\&\left(\begin{array}{l}\texttt{holdsAt}\left(\texttt{extC}\left(o_1,o_2\right),t_2\right)\vee\\\texttt{holdsAt}\left(\texttt{co}\left(o_1,o_2\right),t_2\right)\end{array}\right)\end{aligned}\tag{A.19}$$

$$\texttt{happens}\left(\texttt{makeContactTop}\left(o_1,o_2\right),\left[t_1,t_2\right]\right)\longleftrightarrow$$
$$\begin{aligned}&\texttt{here}\left(\left[o_1,o_2\right],\left[t_1,t_2\right]\right)\wedge\\&\texttt{holdsAt}\left(\texttt{disC}\left(o_1,o_2\right),t_1\right)\wedge\\&\texttt{holdsAt}\left(\texttt{above}\left(o_1,o_2\right),t_1\right)\wedge\\&\left(\begin{array}{l}\texttt{holdsAt}\left(\texttt{extC}\left(o_1,o_2\right),t_2\right)\vee\\\texttt{holdsAt}\left(\texttt{co}\left(o_1,o_2\right),t_2\right)\end{array}\right)\end{aligned}\tag{A.20}$$

$$\texttt{happens}\left(\texttt{makeContactBottom}\left(o_1,o_2\right),\left[t_1,t_2\right]\right)\longleftrightarrow$$
$$\begin{aligned}&\texttt{here}\left(\left[o_1,o_2\right],\left[t_1,t_2\right]\right)\wedge\\&\texttt{holdsAt}\left(\texttt{disC}\left(o_1,o_2\right),t_1\right)\wedge\\&\texttt{holdsAt}\left(\texttt{above}\left(o_2,o_1\right),t_1\right)\wedge\\&\left(\begin{array}{l}\texttt{holdsAt}\left(\texttt{extC}\left(o_1,o_2\right),t_2\right)\vee\\\texttt{holdsAt}\left(\texttt{co}\left(o_1,o_2\right),t_2\right)\end{array}\right)\end{aligned}\tag{A.21}$$

$$\texttt{happens}\left(\texttt{breakContactRight}\left(o_1,o_2\right),\left[t_1,t_2\right]\right)\longleftrightarrow$$
$$\begin{aligned}&\texttt{here}\left(\left[o_1,o_2\right],\left[t_1,t_2\right]\right)\wedge\\&\left(\begin{array}{l}\texttt{holdsAt}\left(\texttt{extC}\left(o_1,o_2\right),t_1\right)\vee\\\texttt{holdsAt}\left(\texttt{co}\left(o_1,o_2\right),t_1\right)\end{array}\right)\wedge\\&\texttt{holdsAt}\left(\texttt{disC}\left(o_1,o_2\right),t_2\right)\wedge\\&\texttt{holdsAt}\left(\texttt{left}\left(o_1,o_2\right),t_2\right)\end{aligned}\tag{A.22}$$

$$\texttt{happens}\left(\texttt{breakContactLeft}\left(o_1,o_2\right),\left[t_1,t_2\right]\right)\longleftrightarrow$$
$$\begin{aligned}&\texttt{here}\left(\left[o_1,o_2\right],\left[t_1,t_2\right]\right)\wedge\\&\left(\begin{array}{l}\texttt{holdsAt}\left(\texttt{extC}\left(o_1,o_2\right),t_1\right)\vee\\\texttt{holdsAt}\left(\texttt{co}\left(o_1,o_2\right),t_1\right)\end{array}\right)\wedge\\&\texttt{holdsAt}\left(\texttt{disC}\left(o_1,o_2\right),t_2\right)\wedge\\&\texttt{holdsAt}\left(\texttt{left}\left(o_2,o_1\right),t_2\right)\end{aligned}\tag{A.23}$$

$$\texttt{happens}\left(\texttt{breakContactRight}\left(o_1,o_2\right),\left[t_1,t_2\right]\right)\longleftrightarrow$$
$$\begin{aligned}&\texttt{here}\left(\left[o_1,o_2\right],\left[t_1,t_2\right]\right)\wedge\\&\left(\begin{array}{l}\texttt{holdsAt}\left(\texttt{extC}\left(o_1,o_2\right),t_1\right)\vee\\\texttt{holdsAt}\left(\texttt{co}\left(o_1,o_2\right),t_1\right)\end{array}\right)\wedge\\&\texttt{holdsAt}\left(\texttt{disC}\left(o_1,o_2\right),t_2\right)\wedge\\&\texttt{holdsAt}\left(\texttt{above}\left(o_1,o_2\right),t_2\right)\end{aligned}\tag{A.24}$$

$$\texttt{happens}\left(\texttt{breakContactBottom}\left(o_1,o_2\right),\left[t_1,t_2\right]\right)\longleftrightarrow$$
$$\begin{aligned}&\texttt{here}\left(\left[o_1,o_2\right],\left[t_1,t_2\right]\right)\wedge\\&\left(\begin{array}{l}\texttt{holdsAt}\left(\texttt{extC}\left(o_1,o_2\right),t_1\right)\vee\\\texttt{holdsAt}\left(\texttt{co}\left(o_1,o_2\right),t_1\right)\end{array}\right)\wedge\\&\texttt{holdsAt}\left(\texttt{disC}\left(o_1,o_2\right),t_2\right)\wedge\\&\texttt{holdsAt}\left(\texttt{above}\left(o_2,o_1\right),t_2\right)\end{aligned}\tag{A.25}$$

Appendix B

Model Equation Derivations

Chapter four presented a series of models, each model determining whether a composite event type exists by assigning a significance measure value, V to the composite event type based on input evidence in favour and against the existence of the composite event type. Each piece of evidence being provided to the model as the system encounters that evidence.

Four of the models introduce equations that require a more involved derivation than could be presented in chapter four, as doing so would spoil the flow of the general description of those models. Those derivations are presented here instead.

B.1 The Count Only Model

The Count Only model provides a significance measure that is based on a frequentist probability that the two events that make-up the composite event type in question are part of the same event type. The formula being derived compares the number of observed instances of a composite event type ($|T_{1,2}|$) with the total number of observed instances of its component event types ($|T_1|$ and $|T_2|$) and is shown in equation B.1.

$$V = \frac{2\,|T_{1,2}|}{|T_1| + |T_2|} \tag{B.1}$$

Consider the undirected graph $G\,(E, A)$ where each vertex in the set E corresponds to an observed event instance and each edge in the set A corresponds to a pairing of event instances created by the association module of the system. Let atomic event types be defined as a subset of the vertices of the graph $T_1 \subseteq E$ and composite event types be defined as a subset of edges of the graph $T_{1,2} \subseteq A$. Let there be a set T, the union of all possible atomic event types and all possible composite event types. Let there be two

functions, a function that maps each vertex to an event type, as defined in equation B.2, and a function that maps each edge to an event type, as defined in equation B.3.

$$\texttt{type} : e \in E \rightarrow T \in \mathbb{T} \tag{B.2}$$

$$\texttt{type} : (e_1, e_2) \in A \rightarrow T \in \mathbb{T} \tag{B.3}$$

The association module applies some constraints on the type of edges that can exist which are expressed in equations B.4 and B.5. The first constraint states that no edge can exist between any two vertices that share the same type. The second constraint states that each vertex may not be connected to more than one vertex of the same type.

$$\forall (e_1, e_2) \in A, \texttt{type}(e_1) \neq \texttt{type}(e_2) \tag{B.4}$$

$$\forall (e_1, e_2) \in A, \nexists (e_1, e_3) \in A, \texttt{type}(e_2) = \texttt{type}(e_3) \tag{B.5}$$

From these definitions and constraints, an inequality can be shown to hold that relates the size of T_1 to the size of $T_{1,2}$.

Proposition: $|T_{1,2}| \leq |T_1|$

Proof. Assume for the purposes of obtaining a contradiction, that the converse inequality is possible, i.e. $|T_{1,2}| > |T_1|$. This states that there are more edges between the vertex sets T_1 and T_2 than vertices, through the pigeonhole principle, there must exist a vertex in T_1 with at least two edges that each connect to a vertex in T_2. A contradiction occurs as the second constraint stated in equation B.5 forbids any one node from being connected to more than one vertex in T_2. This proves that $|T_{1,2}| \not> |T_1|$ holds. In order to prove that both $|T_{1,2}| = |T_1|$ and $|T_{1,2}| < |T_1|$ are possible, it suffices to provide a single case of each case. Consider the case where there is one vertex in T_1, one vertex in T_2 and $T_{1,2}$ contains a single edge connecting the two mentioned vertices together, clearly $|T_{1,2}| = 1$ and $|T_1| = 1$ and so $|T_{1,2}| = |T_1|$. Consider the same situation but add a single vertex to T_1 that is not connected to any other vertex, now $|T_{1,2}| = 1$ and $|T_1| = 2$ and so $|T_{1,2}| < |T_1|$. Therefore, as $|T_{1,2}| = |T_1|$ can hold, $|T_{1,2}| < |T_1|$ can hold and $|T_{1,2}| \not> |T_1|$ holds then $|T_{1,2}| \leq |T_1|$ holds. □

From this inequality, as it is impossible to get set sizes that are negative, a simple re-arrangement of the inequality yields the inequality expressed in equation B.6.

$$0 \leq \frac{|T_{1,2}|}{|T_1|} \leq 1 \tag{B.6}$$

This rearrangement assumes that $|T_1| > 0$, but this is acceptable as there is no practical need to define the significance measure for event types that have never been observed. Besides, as $|T_{1,2}|$ will always be zero if $|T_1|$ is zero

(as any edge in $T_{1,2}$ must be incident on a vertex in T_1), which means that $\frac{|T_{1,2}|}{|T_1|} = \frac{0}{0}$ and so is meaningless anyway. The ratio of this further inequality can be said to be the probability that a vertex in T_1 is connected to a vertex in T_2, or conversely, the probability that an event instance of type T_1 is paired with an event instance of type T_2.

It is not sufficient to just calculate the probability that an event instance of type T_1 is paired with an event instance of type T_2; the significance measure must also refer to the probability that an event instance of type T_2 is paired with an event instance of type T_1. These two probabilities can be different, as the number of instances in each set can be different. In order to incorporate both probabilities within a single measure of significance, the union of both sets of event instances is used instead of using one set or the other. In doing this, an issue that arises is that while both vertices of an edge are being counted independently, the edge itself is being counted once. This issue is dealt with by counting each edge twice, once for each vertex the edge is connected to. The resultant probability, shown in equation B.7, is used as the significance measure for the Count Only model.

$$V = \frac{2\,|T_{1,2}|}{|T_1 \cup T_2|} = \frac{2\,|T_{1,2}|}{|T_1| + |T_2|} \tag{B.7}$$

B.2 The Iterative Acquire-Extinguish Model

The derivation of the Iterative Acquire-Extinguish significance value functions is mostly straightforward algebraic manipulation, though it is quite lengthy. Instead of showing each individual step, the focus will be on the important steps and those steps that are less obvious. Note that in this section, to abstract away from the model, the derivations will deal in terms of x and y values of a function.

The overall idea for turning an absolute function into an iterative one is based on the assumption that the absolute function is a one-to-one relation. While this assumption is only true for a sub-set of all functions, all the functions concerned satisfy this. As a function is one-to-one, given any y value the x value can be given exactly. Therefore, given the y value, and the knowledge of needing to change the x value by one, the change in the y value can be calculated.

The derivation does not use the differential of the functions involved because differentiation assumes that the change in x is infinitesimal, whereas the iterative function assumes that the change in x is integral. To highlight this distinction, the change in x and y will be written as Δx and Δy respectively rather than δx and δy respectively.

The function for positive evidence will be dealt with first, followed by the function for negative evidence.

B.2.1 The Positive Evidence Function

The absolute function for positive evidence is the logistic function, shown in equation B.8.

$$y = \frac{1}{1 + e^{-k_1 x}} \tag{B.8}$$

There are two variants of this equation that are used later, shown in equations B.9 and B.10.

$$e^{-k_1 x} = \frac{1}{y} - 1 \tag{B.9}$$

$$e^{k_1 x} = \frac{y}{1 - y} \tag{B.10}$$

In order to calculate the iterative change, first the relationship between iterations must be stated for both x and y values. This is expressed in equations B.11 and B.12.

$$x_{n+1} = x_n + \Delta x \tag{B.11}$$

$$y_{n+1} = y_n + \Delta y \tag{B.12}$$

The $n+1^{\text{th}}$ value of y can be calculated using the absolute formula of B.8, as shown in B.13. This can be combined with B.15 to give an equation for Δy as is done in B.14 and B.15.

$$y_{n+1} = \frac{1}{1 + e^{-k_1 x_{n+1}}} \tag{B.13}$$

$$y_n + \Delta y = \frac{1}{1 + e^{-k_1 (x_n + \Delta x)}} \tag{B.14}$$

$$\Delta y = \frac{1}{1 + e^{-(k_1 x_n + k_1 \Delta x)}} - y_n \tag{B.15}$$

The n^{th} value of y can also be calculated using the absolute formula of B.8. This can then be substituted for the y_n term of B.15 as shown in equation B.16. Equations B.17 through to B.20 are then stages in the re-arrangement of B.16 to allow for further substitutions to take place with the goal of eliminating all the x_n terms.

$$\Delta y = \frac{1}{1 + e^{-(k_1 x_n + k_1 \Delta x)}} - \frac{1}{1 + e^{-k_1 x_n}} \tag{B.16}$$

$$\Delta y = \frac{e^{k_1 x_n} e^{k_1 \Delta x}}{1 + e^{k_1 x_n} e^{k_1 \Delta x}} - \frac{e^{k_1 x_n}}{1 + e^{k_1 x_n}} \tag{B.17}$$

$$\Delta y = \frac{e^{k_1 x_n} e^{k_1 \Delta x} \left(e^{k_1 x_n} + 1\right) - e^{k_1 x_n} \left(e^{k_1 x_n} e^{k_1 \Delta x} + 1\right)}{\left(e^{k_1 x_n} e^{k_1 \Delta x} + 1\right) \left(e^{k_1 x_n} + 1\right)} \tag{B.18}$$

$$\Delta y = \frac{e^{k_1 x_n} \left(e^{k_1 \Delta x} - 1\right)}{e^{k_1 x_n} \left(e^{k_1 x_n} e^{k_1 \Delta x} + e^{k_1 \Delta x} + 1 + e^{-k_1 x_n}\right)} \tag{B.19}$$

$$\Delta y = \frac{e^{k_1 \Delta x} - 1}{e^{k_1 x_n} e^{k_1 \Delta x} + e^{k_1 \Delta x} + 1 + e^{-k_1 x_n}} \tag{B.20}$$

To eliminate the x_n terms, the re-arrangements of equation B.8 – equations B.9 and B.10 – can be substituted into B.20. This is shown in B.21. The remainder of the derivation, B.22 and B.23, are stages in the simplification to arrive at the final function of equation B.24.

$$\Delta y = \frac{e^{k_1 \Delta x} - 1}{\left(\frac{y_n}{1-y_n}\right) e^{k_1 \Delta x} + e^{k_1 \Delta x} + 1 + \left(\frac{1}{y_n} - 1\right)} \tag{B.21}$$

$$\Delta y = \frac{e^{k_1 \Delta x} - 1}{e^{k_1 \Delta x} \left(\frac{1}{1-y_n}\right) + \frac{1}{y_n}} \tag{B.22}$$

$$\Delta y = \frac{y_n \left(e^{k_1 \Delta x} - 1\right) (1 - y_n)}{y_n e^{k_1 \Delta x} - y_n + 1} \tag{B.23}$$

$$\Delta y = \frac{(1 - y_n)\, e^{k_1 \Delta x} + y_n - 1}{e^{k_1 \Delta x} + \frac{1}{y_n} - 1} \tag{B.24}$$

B.2.2 The Negative Evidence Function

The negative evidence function follows the same lines as the positive evidence function, but is a lot simpler due to the absolute function that the iterative function is based on being a lot simpler. The absolute function is shown in equation B.25 with its rearrangement in B.26.

$$y = -k_2 x \tag{B.25}$$

$$x = \frac{-y}{k_2} \tag{B.26}$$

The $n + 1^{\text{th}}$ value of y is expressed in equation B.27. The substitution of B.12 is applied in equation B.28 and rearranged for Δy in B.29. Equation B.30 eliminates x_n by substituting in B.26. Through some straightforward rearrangements, the final formula is arrived at and is shown in equation B.31.

$$y_{n+1} = -k_2 x_{n+1} \tag{B.27}$$

$$y_n + \Delta y = -k_2 \left(x_n + \Delta x\right) \tag{B.28}$$

$$\Delta y = -k_2 \left(x_n + \Delta x\right) - y_n \tag{B.29}$$

$$\Delta y = -k_2 \left(\frac{-y_n}{k_2} + \Delta x\right) - y_n \tag{B.30}$$

$$\Delta y = -k_2 \Delta x \tag{B.31}$$

B.3 The Temporal Model

The Temporal model is based upon the observation that the log-normal probability distribution function from statistics has a similar shape to that of the inter-stimulus interval curve of classical conditioning. Equation B.32 expresses the log-normal function, where x and y are the horizontal and vertical variables, μ is a parameter that influences where the centre of the curve lies and σ is a parameter that influences the width of the curve[1] In accordance with the curve shape observation, the variable x curve can be seen to correspond to the inter-stimulus interval.

$$y = \frac{e^{\frac{-(\ln x - \mu)^2}{2\sigma^2}}}{x\sigma\sqrt{\frac{\pi}{2}}} \tag{B.32}$$

The research regarding the inter-stimulus interval, as presented in chapter two, stated that the inter-stimulus interval curve is influenced by whether there is any overlap between the two events. There are three common configurations talked about in the classical conditioning literature. The first configuration is where the conditioned stimulus terminates at the same time as the unconditioned stimulus; the second configuration is where the conditioned stimulus terminates at the same time as the unconditioned stimulus begins and the third configuration is where the conditioned stimulus terminates some time before the unconditioned stimulus begins. The larger the distance between the end times of the events becomes the later-starting and narrower the inter-stimulus curve becomes.

In order to replicate this effect between the end times of the events, the measure needs to be codified into a single value ϕ. While it could be thought that simply taking the difference between the two end times would suffice, this does not take into account the relative size of the overlap in relation to the length of the second event. Therefore the value needs to be normalised by the second event length to lie in the range $0 \leq \phi \leq 1$. However, another issue is that when there is a gap between the two events, it is more appropriate to normalise to the size of the moving window instead, as the basis for the window size is as a cut-off point for inter-stimulus interval.

Therefore, to take into account the need to have a single measure that is normalised against two different scales leads to the following solution: The case where the first event terminates as the second begins is defined to lie at 0.5. Where there is an overlap, we normalise the overlap of the two events against the total size of the second event to lie between 0 and 0.5, with 0 if there is no overlap. This normalised overlap is then subtracted from the 0.5 of the case where there is no overlap or gap to define the value ϕ between 0 and 0.5. Where there is a gap between the two events, the size of the gap is normalised against the size of the size of the window to again lie between

[1]The variable descriptions here are describing the variables independently of the function's use within statistics.

0 and 0.5 with 0 if there is no gap. This is then added to the 0.5 of the case where there is no overlap or gap to define the value ϕ between 0.5 and 1. This entire definition of ϕ is expressed in equation B.33. In equation B.33, $t_{E,1}$ denotes the end time of the first event, $t_{S,2}$ is the start time of the second event, $t_{E,2}$ is the end time of the second event, and W represents the size of the moving window.

$$\phi = \frac{1}{2} - \frac{1}{2}\max\left(0, \frac{t_{E,1} - t_{S,2}}{t_{E,2} - t_{S,2}}\right) + \frac{1}{2}\max\left(0, \frac{t_{S,2} - t_{E,1}}{W}\right) \tag{B.33}$$

The inter-stimulus interval is defined by the difference $t_{S,2} - t_{S,1}$ where $t_{S,1}$ is the start time of the first event. However, this is not the quantity that is used in the final curve equation. It was earlier noted that the longer the difference between the end points becomes the larger the later the curve starts to rise. The best method that was found to include this effect was to alter the input inter-stimulus interval such that some multiple s of the ϕ value is subtracted from the inter-stimulus interval, meaning that the larger ϕ is the later the curve starts, so modelling the observation. The log-normal curve is however undefined for negative values of x – this is rectified by setting the result to zero if the subtraction is negative. All this is expressed in equation B.34. Through experimentation, it was determined that an s value of 2 gave a good approximation of the relative starts of the curve based on the observed data. This is reflected in equation B.35.

$$\chi = \max\left(0, \left(B_S - A_S - s\phi\right)\right) \tag{B.34}$$

$$\chi = \max\left(0, \left(B_S - A_S - 2\phi\right)\right) \tag{B.35}$$

These two values are then fed into a modified version of the log-normal curve shown in equation B.36. The value χ has taken the place of x as χ represents the inter-stimulus interval. The value $(2 + \phi)$ has taken place of the σ parameter to allow the difference between the ends times of the events to influence the width of the curve, in accordance with observation. It was found through experimentation that varying the σ parameter between 2 and 3 gave a better difference in the width of the curve. The μ parameter was set to 1 as although each curve needed to start later, each curve needed have a rapid increase when it does start. The final change to the function, the $2\,(2 - \phi)$ section is a scaling factor allowing the ϕ value to influence the overall peak size of the curve. It was found again through experimentation that varying the value between 2 and 4 achieved the best correspondence to the observed curve.

$$Z = \frac{2\,(2 - \phi)\, e^{\frac{-2(\ln(\chi)-1)^2}{(2+\phi)^2}}}{\chi\,(2 + \phi)\,\sqrt{\frac{\pi}{2}}} \tag{B.36}$$

B.4 The Reacquiring Model

The change in the Reacquiring model is multiplying the absolute positive evidence equation by a value that is also has an asymptote at a y value of one. This secondary asymptotic function would increase with the number of positive pieces of evidence but not decrease with the number of negative pieces of evidence. The resultant absolute function is then taken through the same steps as done with the derivation iterative positive evidence function to arrive at a new iterative version that can be influenced by the secondary asymptote. As the derivations are so similar, the derivation presented below is shortened, showing only how the asymptote value propagates through the derivation.

The derivation begins with the absolute positive evidence function that has been multiplied with the value a, representing the secondary asymptote as shown in equation B.37.

$$y = \frac{a}{1 + e^{-k_1 x}} \tag{B.37}$$

Two re-arranged versions of equation B.37 are shown in equations B.38 and B.40.

$$e^{-k_1 x} = \frac{a}{y} - 1 \tag{B.38}$$

$$e^{k_1 x} = \frac{y}{a - y} \tag{B.39}$$

We then combine B.37 with B.12 to give the $n + 1^{\text{th}}$ Δy value as shown in B.40. The value of y_n is the substituted with equation B.37 to give equation B.41 which is then re-arranged to give equation B.42.

$$\Delta y = \frac{a}{1 + e^{-(k_1 x_n + k_1 \Delta x)}} - y_n \tag{B.40}$$

$$\Delta y = \frac{a}{1 + e^{-(k_1 x_n + k_1 \Delta x)}} - \frac{a}{1 + e^{-k_1 x_n}} \tag{B.41}$$

$$\Delta y = \frac{a\left(e^{k_1 \Delta x} - 1\right)}{e^{k_1 x_n} e^{k_1 \Delta x} + e^{k_1 \Delta x} + 1 + e^{-k_1 x_n}} \tag{B.42}$$

In order to eliminate the x_n terms, equations B.38 and B.39 are substituted into B.42 to give equation B.43. This is then rearranged to give the intermediate equation B.44 and the final iterative function shown in equation B.45.

$$\Delta y = \frac{a\left(e^{k_1 \Delta x} - 1\right)}{\left(\frac{y_n}{a - y_n}\right) e^{k_1 \Delta x} + e^{k_1 \Delta x} + 1 + \left(\frac{a}{y_n} - 1\right)} \tag{B.43}$$

$$\Delta y = \frac{e^{k_1 \Delta x} - 1}{e^{k_1 \Delta x}\left(\frac{1}{a - y_n}\right) + \frac{1}{y_n}} \tag{B.44}$$

$$\Delta y = \frac{(a - y_n)\, e^{k_1 \Delta x} + y_n - a}{e^{k_1 \Delta x} + \frac{a}{y_n} - 1} \tag{B.45}$$

The varying asymptote value itself is based upon the absolute count of pieces of positive evidence. The function that is used for the asymptote is a variant of the simplest equation that has an asymptote at a y value of one. The normal version of the function is shown in equation B.46 and the variant is given in equation B.47. In the equations, a is the asymptote value used in the main positive evidence equation and ϵ^+ is the absolute count of positive pieces of evidence provided to the model. In equation B.47, a constant k_3 is introduced that allows control over how quickly the asymptote of one is approached. The whole equation was then squared as it was found through experimentation that by doing this, the equation tails off in a manner that gives a greater scope for later reacquisition phases to noticeably demonstrate a faster rate of acquisition.

$$a = \frac{\epsilon^+}{\epsilon^+ + 1} \tag{B.46}$$

$$a = \left(\frac{\epsilon^+}{\epsilon^+ + k_3}\right)^2 \tag{B.47}$$

However, the equation of B.48 has an undue influence over the first acquisition phase; this is because the value of reacquisition asymptote in this equation is too low. In order to counter this, a minimum asymptote value k_4 is introduced, and the changing asymptote is allowed to only vary between k_4 and one, giving a constraint to the asymptote value of $k_4 \leq a < 1$. This is achieved in the manner shown in equation B.48, which is the actual function used to calculate the asymptote.

$$a = k_4 + (1 - k_4)\left(\frac{\epsilon^+}{\epsilon^+ + k_3}\right)^2 \tag{B.48}$$

Appendix C

Proxy Ground Truth Gantt Charts

Chapter five introduced three learning scenarios which form the environments in which the system discussed in chapter four is tested. Each of the scenarios had a proxy ground truth created to compare against the results produced by the system. The proxy ground truth was created through a combination of restricted human judgement stages and deterministic stages. The first human judgement stage created key-frames for the scenarios, the results of which were included in chapter five. The second human judgement stage involved creating Gantt charts for each of the scenarios, giving the expected length of each atom event that was produced as a consequence of the first human judgement stage. Due to the space taken up by the Gantt charts these are presented in this appendix.

The first scenario – the throwing scenario – is shown in figure C.1. The second scenario – the rotating scenario – is shown in figure C.2. The remainder of the charts, shown in figures C.3 through to C.14, show the twelve different sequences of the third scenario – the colliding scenario. The separate sequences of the colliding scenario are labelled with the letters A to L which correspond to the sequences shown in figures 5.10 and 5.11.

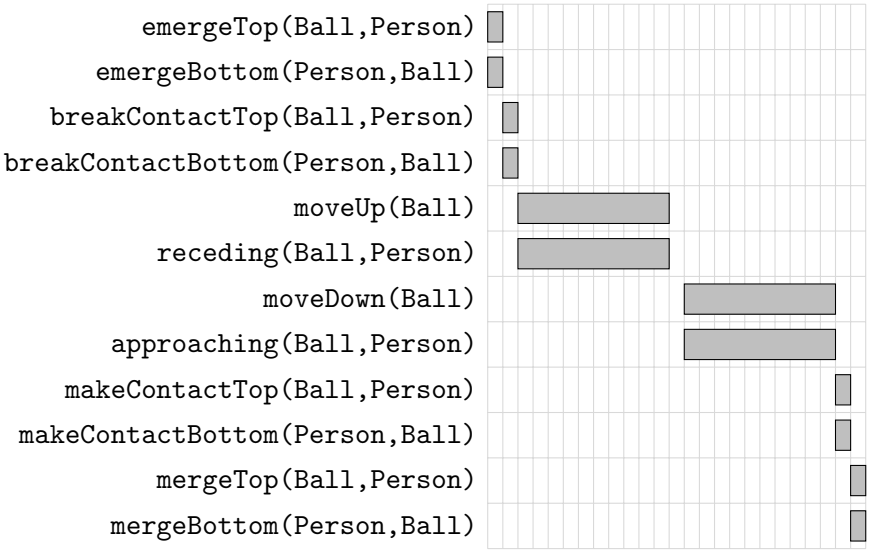

Figure C.1: A Gantt chart showing the temporal relationships of all the atomic event instances of the throwing scenario.

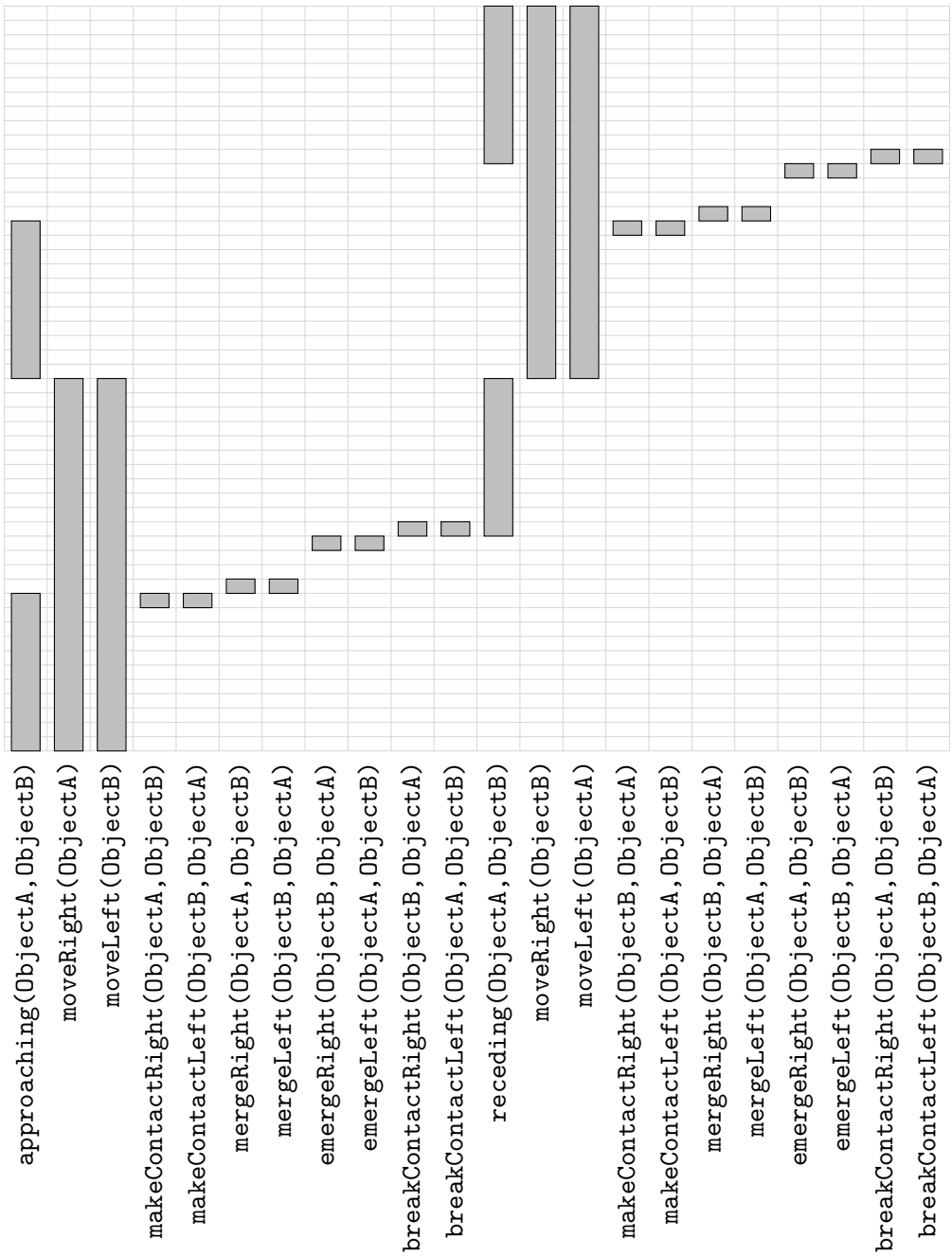

Figure C.2: A Gantt chart showing the temporal relationships of all the atomic event instances of the rotating scenario.

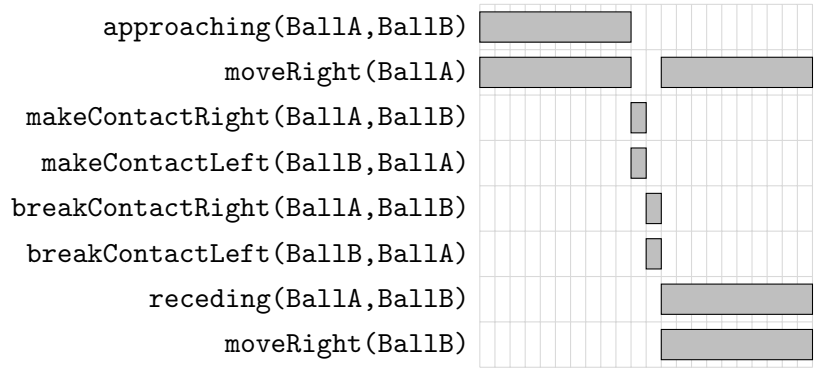

Figure C.3: A Gantt chart showing the temporal relationships of the atomic event instances for sequence A of the colliding scenario.

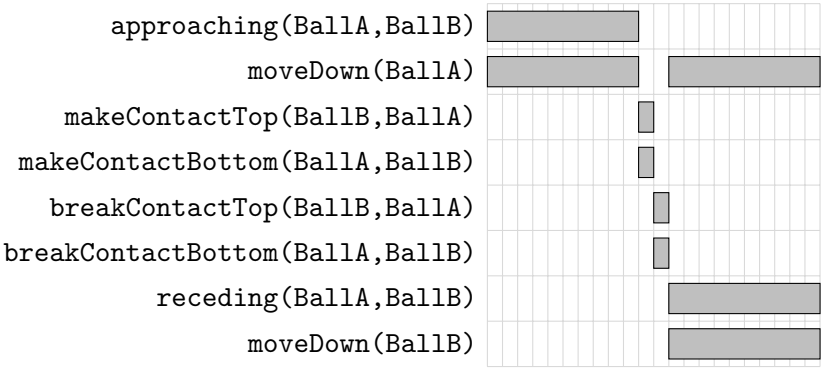

Figure C.4: A Gantt chart showing the temporal relationships of the atomic event instances for sequence B of the colliding scenario.

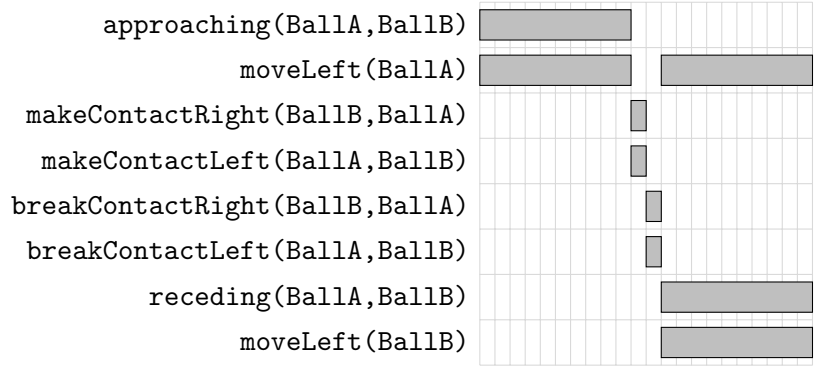

Figure C.5: A Gantt chart showing the temporal relationships of the atomic event instances for sequence C of the colliding scenario.

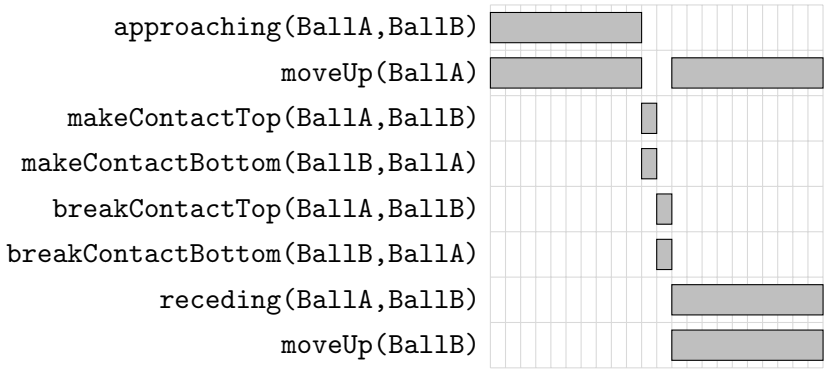

Figure C.6: A Gantt chart showing the temporal relationships of the atomic event instances for sequence D of the colliding scenario.

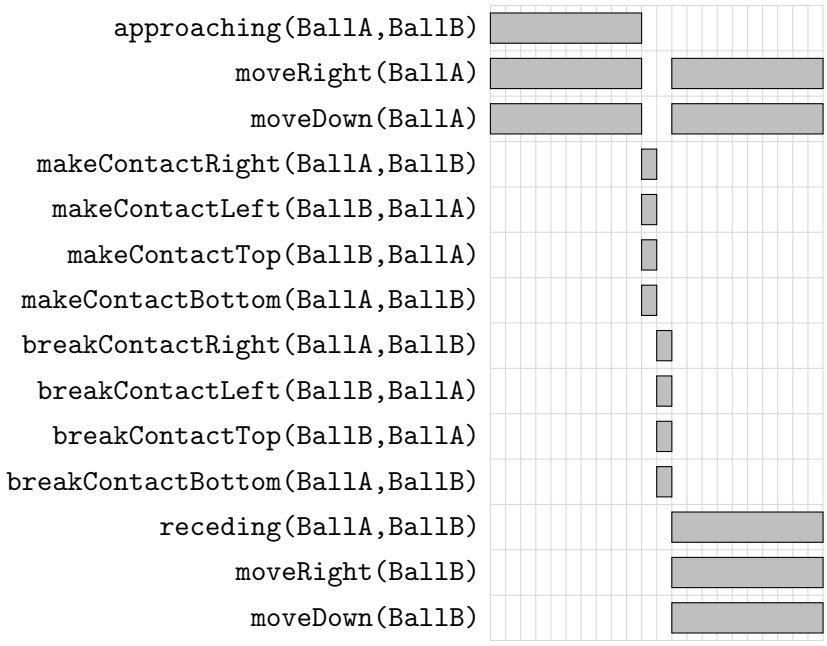

Figure C.7: A Gantt chart showing the temporal relationships of the atomic event instances for sequence E of the colliding scenario.

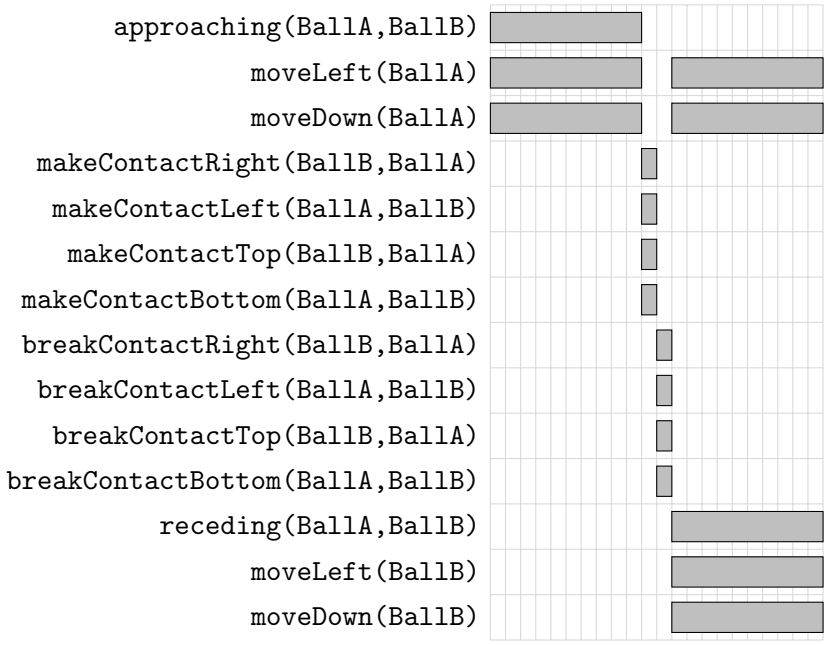

Figure C.8: A Gantt chart showing the temporal relationships of the atomic event instances for sequence F of the colliding scenario.

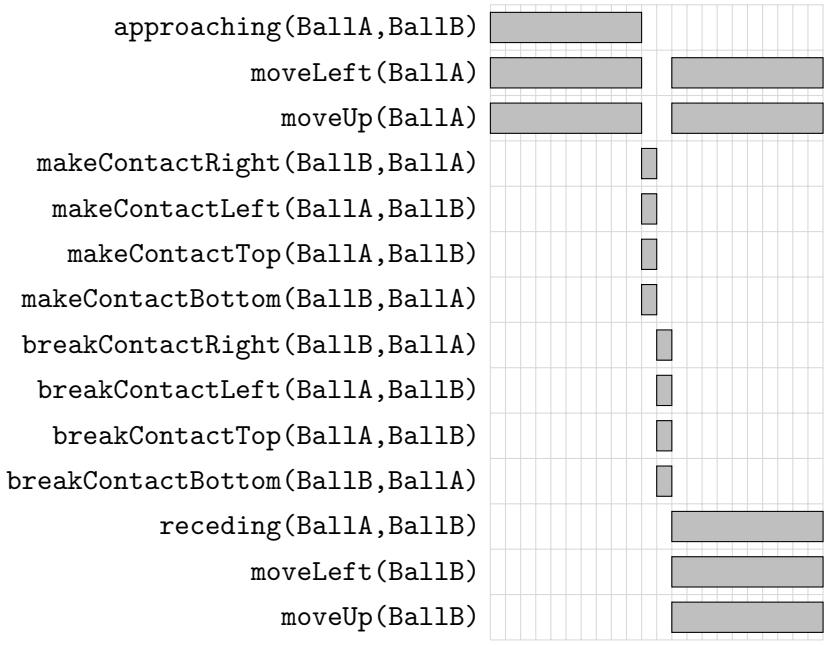

Figure C.9: A Gantt chart showing the temporal relationships of the atomic event instances for sequence G of the colliding scenario.

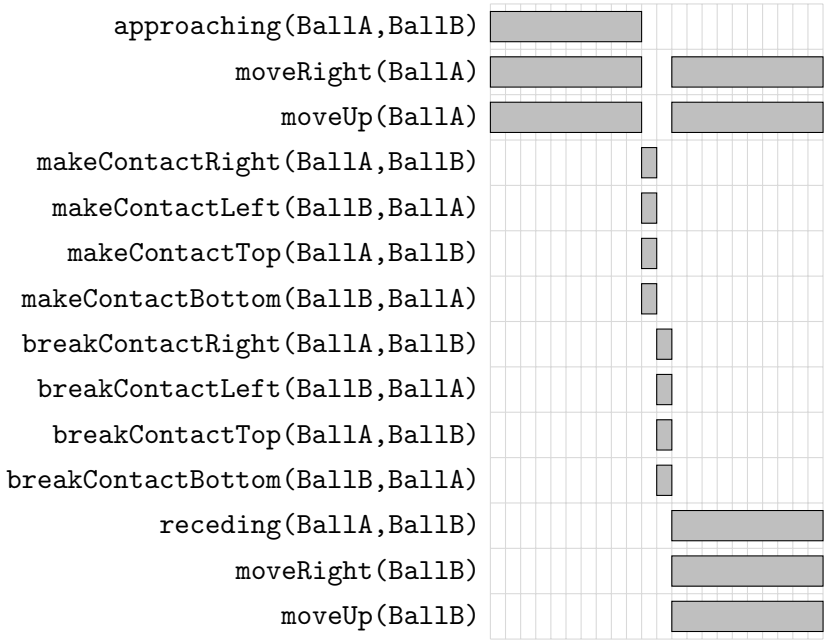

Figure C.10: A Gantt chart showing the temporal relationships of the atomic event instances for sequence H of the colliding scenario.

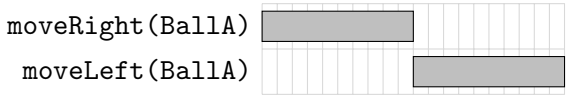

Figure C.11: A Gantt chart showing the temporal relationships of the atomic event instances for sequence I of the colliding scenario.

Figure C.12: A Gantt chart showing the temporal relationships of the atomic event instances for sequence J of the colliding scenario.

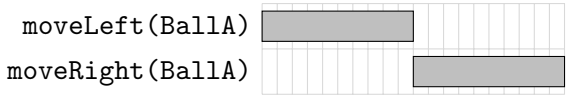

Figure C.13: A Gantt chart showing the temporal relationships of the atomic event instances for sequence K of the colliding scenario.

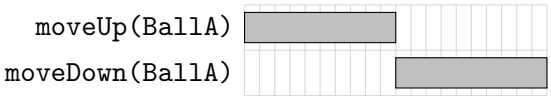

Figure C.14: A Gantt chart showing the temporal relationships of the atomic event instances for sequence L of the colliding scenario.

Appendix D

System Settings

This appendix provides a list of each constant within the system, including model settings, noting the value that was used. The symbols noted in each entry are the symbol the constant was assigned in chapter four. The Count Only model has no settings and so is not listed.

System-wide Constants

- Window size (W): 6 frames
- Upper Significance Threshold (τ_{upper}): 0.96
- Lower Significance Threshold (τ_{lower}): 0.01

Model Constants

- Fixed Increment
  - Positive Increment (f^+): 0.01
- Symmetrical Fixed Increment
  - Positive Increment (f^+): 0.01
  - Negative Increment (f^-): -0.01
- Absolute Acquire-Extinguish
  - Reinforcement Learning Rate (k_1): 0.5
  - Non-Reinforcement Learning Rate (k_2): 1.0

- Iterative Acquire-Extinguish
  - Reinforcement Learning Rate (k_1): 0.5
  - Non-Reinforcement Learning Rate (k_2): 1.0
- Temporal
  - Reinforcement Learning Rate (k_1): 0.5
  - Non-Reinforcement Learning Rate (k_2): 1.0
- Reacquiring
  - Reinforcement Learning Rate (k_1): 0.5
  - Non-Reinforcement Learning Rate (k_2): 1.0
  - Reacquisition Rate (k_3): 2.0
  - Reacquisition Base (k_4): 0.8
- Blocking
  - Reinforcement Learning Rate (k_1): 0.5
  - Non-Reinforcement Learning Rate (k_2): 1.0
  - Reacquisition Rate (k_3): 2.0
  - Reacquisition Base (k_4): 0.8
- Inhibition
  - Reinforcement Learning Rate (k_1): 0.5
  - Non-Reinforcement Learning Rate (k_2): 1.0
  - Reacquisition Rate (k_3): 2.0
  - Reacquisition Base (k_4): 0.8
  - Inhibition Increase Rate (k_5): 1.0
  - Inhibition Decrease Rate (k_6): 1.0
  - Inhibition Rule Removal Threshold (τ_I): 0.001
- Pre-Exposure
  - Reinforcement Learning Rate (k_1): 0.5
  - Non-Reinforcement Learning Rate (k_2): 1.0
  - Reacquisition Rate (k_3): 2.0
  - Reacquisition Base (k_4): 0.8
  - Inhibition Increase Rate (k_5): 1.0
  - Inhibition Decrease Rate (k_6): 1.0
  - Inhibition Rule Removal Threshold (τ_I): 0.001

References

Allen, James F. (November 1983). "Maintaining Knowledge about Temporal Intervals". *Communications of the ACM*, **26**(11), pages 832–843.

Alonso, Eduardo (2002). "Safe learning agents? Logic-based agents!" In Barley, Mike & Guesgen, Hans W. (editors), *Proceedings of the 2002 AAAI Spring Symposium: Safe Learning Agents*, volume Technical Report SS-02-07, pages 1–4. AAAI Press, Menlo Park, California.

Alonso, Eduardo, Mondragón, Esther & Fernández, Alberto (October 2012). "A Java Simulator of Rescorla and Wagner's Prediction Error Model and Configural Cue Extensions". *Computer Methods and Programs in Biomedicine*, **108**(1), pages 346–355.

Alonso, Eduardo, Mondragón, Esther & Kjäll-Ohlsson, Niclas (3 April 2006). "Pavlovian and Instrumental Q-learning: A Rescorla-Wagner-based Approach to Generalisation in Q-Learning". In Kovacs, Tim & Marshall, James A. R. (editors), *Proceedings of the 2006 Convention on Artificial Intelligence and the Simulation of Behaviour: Adaptation in Artificial and Biological Systems*, volume 1 of 3, pages 23–29. Society for the Study of Artificial Intelligence and the Simulation of Behaviour, Bristol, U.K.

Alonso, Eduardo & Schmajuk, Nestor A. (September 2012). "Special Issue on Computational Models of Classical Conditioning Guest Editors' Introduction". *Learning and Behavior*, **40**(3), pages 231–240.

Anderson, John R. (2000). *Learning and Memory: An Integrated Approach*. John Wiley and Sons, Inc., New York, U.S.A, 2nd edition.

Argall, Brenna D., Chernova, Sonia, Veloso, Manuela & Browning, Brett (May 2009). "A Survey of Robot Learning from Demonstration". *Robotics and Autonomous Systems*, **57**(5), pages 469–483.

Balkenius, Christian & Morén, Jan (1998). "Computational Models of Classical Conditioning: A Comparative Study". Technical Report LUCS 62, Lund University Cognitive Studies, Lund, Sweden.

Bellingham, William P., Gillette-Bellingham, Katy & Kehoe, E. James (June 1985). "Summation and configuration in patterning schedules with the rat and rabbit". *Animal Learning and Behavior*, **13**(2), pages 152–164.

Bellman, Richard (1957). *Dynamic Programming*. Prinston University Press, Prinston, New Jersey, U.S.A.

Bennett, Brandon (24–27 May 1994). "Spatial Reasoning with Propositional Logic". In Doyle, Jon, Sandewall, Erik & Torasson, Pietro (editors), *Principles of Knowledge Representation and Reasoning: Proceedings of the 4th International Conference (KR94)*, pages 51–62. Morgan-Kaufmann, Bonn, Germany.

Bennett, Brandon, Cohn, Anthony G., Wolter, Frank & Zakharyaschev, Michael (November 2002). "Multi-Dimensional Modal Logic as a Framework for Spatio-Temporal Reasoning". *Applied Intelligence*, **17**(3), pages 239–251.

Bennett, Brandon & Galton, Antony P. (March 2004). "A unifying semantics for time and events". *Artificial Intelligence*, **153**(1–2), pages 13–48.

Bennett, Brandon, Magee, Derek R., Cohn, Anthony G. & Hogg, David C. (January 2008). "Enhanced Tracking and Recognition of Moving Objects by Reasoning about Spatio-Temporal Continuity". *Image and Vision Computing*, **26**(1), pages 67–81.

Black, Paul E. (11 April 2005). "Layered Graph". In Black, Paul E. (editor), *Dictionary of Algorithms and Data Structures*, U.S. National Institute of Standards and Technology.
URL: *www.nist.gov/dads/HTML/layeredgraph.html*
[accessed: 26 July 2013]

Brogden, W. J. (September 1939). "Sensory Pre-Conditioning". *Journal of Experimental Psychology*, **25**(4), pages 323–332.

Buşoniu, Lucian, Babuška, Robert & de Schutter, Bart (March 2008). "A Comprehensive Survey of Multiagent Reinforcement Learning". *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, **38**(2), pages 156–172.

Carpenter, Gail A. & Grossberg, Stephen (January 1987). "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine". *Computer Vision, Graphics, and Image Processing*, **37**(1), pages 54–115.

Cayley, George (November 1809). "On Aerial Navigation". *A Journal of Natural Philosophy, Chemistry and The Arts*, **24**(3), pages 164–173.

Chapman, Gretchen B. (September 1991). "Trial Order Affects Cue Interaction in Contingency Judgment". *Journal of Experimental Psychology: Learning, Memory, and Cognition*, **17**(5), pages 837–854.

Chklovski, Timothy (October 2003). "Learner: A System for Acquiring Commonsense Knowledge by Analogy". In Gennari, John, Porter, Bruce & Gil, Yolanda (editors), *Proceedings of the Second International Conference on Knowledge Capture*, pages 4–12. ACM, Sanibel Island, Florida, U.S.A.

Chklovski, Timothy & Gil, Yolanda (9–13 July 2005). "An Analysis of Knowledge Collected from Volunteer Contributors". In Veloso, Manuela & Kambhampati, Subbarao (editors), *Proceedings of the Twentieth National Conference on Artificial Intelligence*, pages 564–570. AAAI, Pittsburgh, Pennsylvania, U.S.A.

Clementini, Eliseo, di Felice, Paolino & van Oosterom, Peter (23–25 June 1993). "A Small Set of Formal Topological Relationships Suitable for End User Interaction". In Abel, David J. & Ooi, Beng Chin (editors), *Proceedings of the Third International Symposium on Advances in Spatial Databases*, volume Lecture Notes In Computer Science 692, pages 277–295. Springer, Singapore.

Cohn, Anthony G., Bennett, Brandon, Gooday, John & Gotts, Nicholas Mark (October 1997). "Qualitative Spatial Representation and Reasoning with the Region Connection Calculus". *Geoinformatica*, **1**(2), pages 275–316.

Colmerauer, Alain, Kanoui, Henry, Roussel, Philippe & Pasero, Robert (1973). "Un système de communication homme-machine en Français". Technical report, Research Group on Artificial Intelligence, Marseille, France.

Colmerauer, Alain & Roussel, Philippe (1996). "The Birth of Prolog". In Bergin, Thomas J. & Gibson, Richard G. (editors), *History of Programming Languages, Volume 2*, pages 37–52. ACM, New York, U.S.A.

Cristani, Marco, Bicego, Manuele & Murino, Vittorio (February 2007). "Audio-Visual Event Recognition in Surveillance Video Sequences". *IEEE Transactions on Multimedia*, **9**(2), pages 257–267.

Cui, Peng, Sun, Li-Feng, Liu, Zih-Qiang & Yang, Shi-Qiang (June 2007). "A Sequential Monte Carlo Approach to Anomaly Detection in Tracking Visual Events". In Baker, Simon, Matas, Jiri, Zabih, Ramin, Kanade, Takeo & Medioni, Gerard (editors), *Proceedings of the Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, IEEE, Minneapolis, Minnesota, USA.

Davis, Ernest (1990). *Representations of Commonsense Knowledge*. Morgan Kaufmann Publishers Inc., San Mateo, California, USA.

Dee, Hannah M. & Hogg, David C. (February 2009). "Navigational Strategies in Behaviour Modelling". *Artificial Intelligence*, **173**(2), pages 329–342.

del Mondo, Géraldine, Stell, John G., Claramut, Christophe & Thibaud, Rémy (June 2010). "A Graph Model for Spatio-Temporal Evolution". *Journal of Universal Computer Science*, **16**(11), pages 1452–1477.

Donegan, Nelson H. & Wagner, Allan R. (1987). "Conditioned Diminution and Facilitaion of the UR: A Sometimes Opponent-Process Interpretation". In Gormezano, Isidore, Prokasy, William F. & Thompson, Richard F. (editors), *Classical Conditioning III*, chapter 13, pages 339–369. Lawrence Erlbaum Associates, Hillsdale, New Jersey, U.S.A.

D'Orazio, T., Leo, M., Spagnolo, P., Nitti, M., Mosca, N. & Distante, A. (May 2009). "A Visual System for Real Time Detection of Goal Events During Soccer Matches". *Computer Vision and Image Understanding*, **113**(5), pages 622–632.

dos Santos, Marcus V., de Brito, Rod C., Park, Ho-Hyun & Santos, Paulo E. (October 2009). "Logic-Based Interpretation of Geometrically Observable Changes Occurring in Dynamic Scenes". *Applied Intelligence*, **31**(2), pages 161–179.

Egenhofer, Max J. (21-23 June 1989). "A Formal Definition of Binary Topological Relationships". In Litwin, Witold & Schek, Hans-Jörg (editors), *Third International Conference on Foundations of Data Organization and Algorithms*, volume Lecture Notes in Computer Science 367, pages 457–472. Springer, Paris, France.

Egenhofer, Max J. (28-30 August 1991). "Reasoning About Binary Topological Relations". In Günther, Oliver & Schek, Hans-Jörg (editors), *Proceedings of the Second International Symposium on Advances in Spatial Databases*, volume Lecture Notes in Computer Science 525, pages 141–160. Springer, Zürich.

Egenhofer, Max J., Sharma, Jayant & Mark, David (30 October – 1 November 1993). "A Critical Comparison of the 4-Intersection and 9-Intersection Models for Spatial Relations: Formal Analysis". In McMaster, Robert B. & Armstrong, Marc P. (editors), *Proceedings of the Eleventh International Symposium On Computer-Assisted Cartography (Autocarto 11)*, pages 1–11. American Congress on Surveying and Mapping, Minneapolis, Minnesota, U.S.A.

Egger, M. David & Miller, Neal E. (August 1962). "Secondary Reinforcement in Rats as a Function of Information Value and Reliability of the Stimulus". *Journal of Experimental Psychology*, **64**(2), pages 97–104.

Epstein, Seymour (December 1967). "Toward a Unified Theory of Anxiety". *Progress in Experimental Personality Research*, **4**, pages 1–89.

Fairbank, Michael & Alonso, Eduardo (January 2011). "The Local Optimality of Reinforcement Learning by Value Gradients, and its Relationship to Policy Gradient Learning". eprint arXiv:1101.0428.

Fairbank, Michael & Alonso, Eduardo (10–15 June 2012). "Value-Gradient Learning". In Abbass, Hussein, Essam, Daryl & Sarker, Ruhul (editors), *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 3062–3069. IEEE, Brisbane, Australia.

Fisher, Ronald A. (1958). *Statistical Methods for Research Workers*. Oliver and Boyd, London, U.K., 13th edition.

Foresti, G. L., Micheloni, C. & Snidaro, L. (August 2004). "Event Classification for Automatic Visual-Based Surveillance of Parking Lots". In Kittler, Josef, Petrou, Maria & Nixon, Mark (editors), *Proceedings of the Seventeenth International Conference on Pattern Recognition*, volume 3, pages 314–317. IEEE Computer Society, IEEE, Cambridge, UK.

Frank, Andrew U. (December 1992). "Qualitative Spatial Reasoning about Distances and Directions in Geographic Space". *Journal of Visual Languages and Computing*, **3**(4), pages 343–371.

Freska, Christian (21–23 September 1992). "Using Orientation Information for Qualitative Spatial Reasoning". In Frank, Andrew U., Campari, Irene & Formentini, Ubaldo (editors), *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, International Conference GIS - From Space to Territory: Theories and Methods of Spatio-Temporal Reasoning*, volume Lecture Notes in Computer Science 639, pages 162–178. Springer, Pisa, Italy.

Furze, Timothy A. & Bennett, Brandon (5 April 2011). "Using the Principles of Classical Conditioning to Learn Event Sequences". In Kazakov, Dimitar & Tsoulas, George (editors), *Proceedings of the Artificial Intelligence and the Simulation of Behaviour (AISB) Conference 2011*, volume Computational Models of Cognitive Development, pages 40–47. Society for the Study of Artificial Intelligence and the Simulation of Behaviour, York, U.K.

Galton, Antony P. (2000). *Qualitative Spatial Change*. Oxford University Press, Oxford, U.K.

Galton, Antony P. (May 2009). "Spatial and Temporal Knowledge Representation". *Earth Science Informatics*, **2**(3), pages 169–187.

Gao, Xiaoying & Sterling, Leon (1 December 1997). "Using Limited Common Sense Knowledge to Guide Knowledge Acquisition for Information Agents". In *Proceedings of the Third Australian Knowledge Acquisition Workshop, in conjunction with the Tenth Australian Joint Conference on Artificial Intelligence*, Perth, Australia.

Gerevini, Alfonso & Nebel, Bernhard (21–26 July 2002). "Qualitative Spatio-Temporal Reasoning with RCC-8 and Allen's Interval Calculus: Computational Complexity". In van Harmelen, Frank (editor), *Proceedings of the 15th European Conference on Artificial Intelligence*, pages 312–316. IOS Press, Lyon, France.

Gosavi, Abhijit (Spring 2009). "Reinforcement Learning: A Tutorial Survey and Recent Advances". *INFORMS Journal on Computing*, **21**(2), pages 178–192.

Grant, David A. & Schipper, Lowell M. (April 1952). "The Acquisition and Extinction of Conditioned Eyelid Responses as a Function of the Percentage of Fixed-Ratio Random Reinforcement". *Journal of Experimental Psychology*, **43**(4), pages 313–320.

Grira, Nizar, Crucianu, Michel & Boujemaa, Nozha (July 2004). "Unsupervised and Semi-supervised Clustering: a Brief Survey". Published as part of "A Review of Machine Learning Techniques for Processing Multimedia Content", A report of the MUSCLE European Network of Excellence.
URL: *satoh-lab.ex.nii.ac.jp/users/grira/papers/BriefSurveyClustering.pdf*
[accessed: 26 July 2013]

Grossberg, Stephen (June 1969). "Embedding Fields: A Theory of Learning with Physiological Implications". *Journal of Mathematical Psychology*, **6**(2), pages 209–239.

Grossberg, Stephen (1974). "Classical and Instrumental Learning by Neural Networks". *Progress in Theoretical Biology*, **3**, pages 51–141.

Grossberg, Stephen (August 1982). "A Psychophysiological Theory of Reinforcement, Drive, Motivation and Attention". *Journal of Theoretical Neurobiology*, **1**(1), pages 286–369.

Grossberg, Stephen & Schmajuk, Nestor A. (1987). "Neural Dynamics of Attentionally Modulated Pavlovian Conditioning: Conditioned Reinforcement, Inhibition and Opponent Processing". *Psychobiology*, **15**(3), pages 195–240.

Grossberg, Stephen & Schmajuk, Nestor A. (1989). "Neural Dynamics of Adaptive Timing and Temporal Discrimination During Associative Learning". *Neural Networks*, **2**(2), pages 79–102.

Guha, Ramanathan V. & Lenat, Douglas B. (1990). "CYC: A Mid-term Report". *Applied Artificial Intelligence: An International Journal*, **5**(1), pages 45–86.

Gupta, Rakesh & Kochenderfer, Mykel J. (25–29 July 2004). "Common Sense Data Acquisition for Indoor Mobile Robots". In Hendler, James A. (editor), *Nineteenth National Conference on Artificial Intelligence*, pages 605–610. AAAI, San Jose, California, U.S.A.

Hadidi, Bobak, Johri, Nikhil, Pantley, Darin, Pradhan, Abhishek & Wang, Felix (12 March 2010). "Automated Knowledge Extraction from Wikipedia". In *University of Illinois at Urbana-Champaign Engineering Open House 2010*, pages 1–6.
URL: *www.acm.uiuc.edu/archives/sigart-l/pdfMANiS52zXR.pdf*
[accessed: 26 July 2013]

Hall, Geoffrey & Pearce, John M. (January 1979). "Latent Inhibition of a CS During CS-US Pairings". *Journal of Experimental Psychology: Animal Behavior Processes*, **5**(1), pages 31–42.

Harlow, Harry F. (January 1949). "The Formation of Learning Sets". *Psychological Review*, **56**(1), pages 51–65.

Havasi, Catherine, Speer, Robert & Alonso, Jason B. (27–29 September 2007). "ConceptNet 3: a Flexible, Multilingual Semantic Network for Common Sense Knowledge". In *Proceedings of Recent Advances in Natural Languages Processing*, Borovets, Bulgaria.

Hebb, Donald O. (1949). *The Organization of Behavior.* Wiley, New York, New York, U.S.A.

Hofstadter, Douglas R. (1979). *Gödel, Escher, Bach: an Eternal Golden Braid.* Penguin, London, U.K.

Hofstadter, Douglas R. (2007). *I Am a Strange Loop.* Basic Books, New York, U.S.A.

Holland, Peter C. (July 1985). "Element Pretraining Influences the Content of Appetitive Serial Compound Conditioning in Rats". *Journal of Experimental Psychology: Animal Behavior Processes*, **11**(3), pages 367–387.

Holland, Peter C. & Rescorla, Robert A. (October 1975). "The Effect of Two Ways of Devaluing the Unconditioned Stimulus after First- and Second-Order Appetitive Conditioning". *Journal of Experimental Psychology: Animal Behavior Processes*, **1**(4), pages 355–363.

Hoogs, Anthony, Rittscher, Jens, Stein, Gees & Schmiederer, John (June 2003). "Video Content Annotation Using Visual Analysis and a Large Semantic Knowledgebase". In Martin, Danielle (editor), *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 2, pages 327–334. IEEE Computer Society, IEEE, Madison, Wisconsin, USA.

Humphreys, Lloyd G. (August 1939). "The Effect of Random Alternation of Reinforcement on the Acquisition and Extinction of Conditioned Eyelid Reactions". *Journal of Experimental Psychology*, **25**(2), pages 141–158.

Hyötyniemi, Heikki (19–25 August 1996). "Turing Machines are Recurrent Neural Networks". In Alander, Jarmo, Honkela, Timo & Jakobsson, Matti

(editors), *Proceedings of STeP'96: Symposium on Artificial Neural Networks*, pages 13–24. Finnish Artificial Intelligence Society, Vaasa, Finland.

Ivanov, Yuri A. & Bobick, Aaron F. (August 2000). "Recognition of Visual activities and interactions by Stochastic Parsing". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(8), pages 852–872.

Kaelbing, Leslie P., Littman, Michael L. & Moore, Andrew W. (May 1996). "Reinforcement Learning: A Survey". *Journal of Artificial Intelligence Research*, **4**, pages 237–285.

Kamin, Leon J. (1969). In Campbell, B. A. & Church, R. M. (editors), *Punishment and aversive behavior*, pages 279–296. Appleton-Century-Crofts, New York.

Kehoe, E. James (October 1988). "A Layered Network Model of Associative Learning: Learning to Learn and Configuration". *Psychological Review*, **95**(4), pages 411–433.

Kehoe, E. James, Gibbs, Charles M., Garcia, Eduardo & Gormezano, Isidore (January 1979). "Associative Transfer and Stimulus Selection in Classical Conditioning of the Rabbit's Nictitating Membrane Response to Serial Compound CSs". *Journal of Experimental Psychology: Animal Behavior Processes*, **5**(1), pages 1–18.

Kehoe, E. James & Holt, Phoebe E. (June 1984). "Transfer Across CS-US Intervals and Sensory Modalities in Classical Conditioning of the Rabbit". *Animal Learning and Behavior*, **12**(2), pages 122–128.

Kehoe, E. James, Schreurs, Bernard G. & Graham, Peita (December 1987). "Temporal Primacy Overrides Prior Training in Serial Compound Conditioning of the Rabbit's Nictitating Membrane Response". *Animal Learning and Behavior*, **15**(4), pages 455–464.

Klein, Stephen B. (1996). *Learning: Principles and Applications*. McGraw-Hill Inc., New York, U.S.A, 3rd edition.

Klopf, A. Harry (1972). "Brain Function and Adaptive Systems - A Heterostatic Theory". Air force cambridge research laboratories special report number 133 (afcrl-72-0164), Defence Technical Information Center Report AD 742259.

Klopf, A. Harry (1988). "A Neuronal Model of Classical Conditioning". *Psychobiology*, **16**(2), pages 85–125.

Kowalski, Robert & Sergot, Marek (March 1986). "A Logic-based Calculus of Events". *New Generation Computing*, **4**(1), pages 67–95.

Kuo, Yen-Ling & Hsu, Jane Yung-Jen (19–22 July 2011). "Resource-Bounded Crowd-Sourcing of Commonsense Knowledge". In *Proceedings of*

the Twenty-Second International Joint Conference on Artificial Intelligence, volume 3, pages 2470–2475. AAAI Press, Barcelona, Spain.

Kuo, Yen-Ling, Shih, Fuming & Hsu, Jane Yung-Jen (23 July 2012). "Contextual Commonsense Knowledge Acquisition from Social Content by Crowd-Sourcing Explanations". In Chen, Yiling (editor), *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence: Human Computation*, AAAI.

Kutlu, Munir Gunes & Schmajuk, Nestor A. (March 2012). "Deactivation and Reactivation of the Inhibitory Power of a Conditioned Inhibitor: Testing the Predictions of an Attentional-Associative Model". *Learning and Behavior*, **40**(1), pages 83–97.

Lenat, Douglas B. (July 1976). *AM: An Artificial Intelligence Approach to Discovery in Mathematics as Heuristic Search*. Ph.D. thesis, Stanford University AI Lab, Stanford, California, U.S.A.

Lenat, Douglas B. (November 1996). "From 2001 to 2001: Common Sense and the Mind of HAL". In Stork, David G. (editor), *Hal's Legacy: 2001's Computer as Dream and Reality*, chapter 9, pages 193–210. MIT Press, Cambridge, Massachusetts, U.S.A.

Lenat, Douglas B. & Guha, Ramanathan V. (1990). *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley, Reading, Massachusetts, U.S.A.

Lenat, Douglas B., Prakash, Mayank & Shepard, Mary (March 1985). "CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks". *AI Magazine*, **6**(4), pages 65–85.

Levesque, Hector, Pirri, Fiora & Reiter, Ray (December 1998). "Foundations for the Situation Calculus". *Electronic Transactions on Artificial Intelligence*, **2**(3–4), pages 159–178.

Liu, Chunxi, Huang, Qingming, Jiang, Shuqiang, Xing, Liyuan, Ye, Qixiang & Gao, Wen (March 2009). "A Framework for Flexible Summarization of Racquet Sports Video Using Multiple Modalities". *Computer Vision and Image Understanding*, **113**(3), pages 415–424.

Liu, Hugo & Singh, Push (October 2004). "ConceptNet – A Practical Commonsense Reasoning Tool-Kit". *BT Technology Journal*, **22**(4), pages 211–226.

Lozano-Peréz, Tomás (February 1983). "Spatial Planning: A Configuration Space Approach". *IEEE Transactions on Computers*, **32**(2), pages 108–120.

Lubow, R. E. & Moore, A. U. (August 1959). "Latent Inhibition: The Effect of Non-Reinforced Pre-Exposure to the Conditional Stimulus". *Journal of Comparative and Physiological Psychology*, **52**(4), pages 415–419.

Luce, R. Duncan (1959). *Induvidual Choice Behavior: A Theoretical Analysis*. John Wiley and Sons, Inc., New York, U.S.A.

Ludvig, Elliot A., Sutton, Richard S. & Kehoe, E. James (September 2012). "Evaluating the TD Model of Classical Conditioning". *Learning and Behavior*, **40**(3), pages 305–319.

Mackintosh, Nicholas J. (July 1975). "A Theory of Attention: Variations in the Associability of Stimuli with Reinforcement". *Psychological Review*, **82**(4), pages 276–298.

Mancilla-Caceres, Juan F. & Amir, Eyal (October 2010). "A Turing Game for Commonsense Knowledge Extraction". In Havasi, Catherine, Lenat, Douglas B. & Van Durme, Benjamin (editors), *Commonsense Knowledge: Papers from the AAAI Fall Symposium*, pages 76–81. AAAI.

Mancilla-Caceres, Juan F. & Amir, Eyal (5–9 September 2011). "Evaluating Commonsense Knowledge with a Computer Game". In Campos, Nicholas, Pedro Graham, Jorge, Joaquim, Nunes, Nuno, Palanque, Philippe & Winckler, Marco (editors), *INTERACT 2011: 13th International Conference on Human-Computer Interaction*, pages 348–355. Springer, Lisbon, Portugal.

Matthews, Brian W. (20 October 1975). "Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme". *Biochimica et Biophysica Acta (BBA) - Protein Structure*, **405**(2), pages 442–451.

Matuszek, Cynthia, Witbrock, Michael, Kahlert, Robert C., Cabral, John, Schneider, Dave & Lenat, Douglas B. (9–13 July 2005). "Searching for Common Sense: Populating Cyc from the Web". In Veloso, Subbarao, Manuela; Kambhampati (editor), *Proceedings of the 20th National Conference on Artificial intelligence*, volume 3, pages 1430–1435. AAAI, Pittsburgh, Pennsylvania, U.S.A.

Matzel, L. D., Schachtman, T. R. & Miller, R. R. (November 1985). "Recovery of an Overshadowed Association Achieved by Extinction of the Overshadowing Stimulus". *Learning and Motivation*, **16**(4), pages 398–412.

McCarthy, John (1959). "Programs with Common Sense". In *Proceedings of the National Physical Laboratory Symposium Number 10: Mechanization of Thought Processes 24-27 Noveber 1958*, volume 1, pages 75–91. Her Majesty's Stationary Office, London, U.K.

McCarthy, John (July 1963). "Situations, Actions, and Causal Laws". Stanford AI Project Memo 2, Stanford University Department of Computer Science, Palo Alto, California, USA.

McCarthy, John (April 1980). "Circumscription – A Form of Non-Monotonic Reasoning". *Artificial Intelligence*, **13**(1–2), pages 27–39.

McCarthy, John (February 1986). "Applications of Circumscription to Formalizing Common Sense Knowledge". *Artificial Intelligence*, **28**(1), pages 89–116.

McCarthy, John & Hayes, Patrick J. (1969). "Some Philosophical Problems from the Standpoint of Artificial Intelligence". In Meltzer, Bernard & Michie, Donald (editors), *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, Edinburgh, UK.

McCulloch, Warren S. & Pitts, Walter (1 December 1943). "A Logical Calculus of the Ideas Immanent in Nervous Activity". *Bulletin of Mathematical Biology*, **5**(4), pages 115–133.

Michie, D. & Chambers, R. A. (1968). "Boxes: An Experiment in Adaptive Control". *Machine Intelligence*, **2**, pages 137–152.

Miller, Ralph R., Barnet, Robert C. & Grahame, Nicholas J. (May 1995). "Assessment of the Rescorla-Wagner Model". *Psychological Bulletin*, **117**(3), pages 363–386.

Miller, Rob & Shanahan, Murray (September 1999). "The Event Calculus in Classical Logic – Alternative Axiomatisations". *Electronic Transactions on Artificial Intelligence*, **3**(Section A), pages 77–105.

Minsky, Marvin L. (1952). "A Neural-Analogue Calculator Based upon a Probability Model of Reinforcement". Technical report, Harvard University Psychological Laboratories, Cambridge, Massachusetts, U.S.A.

Minsky, Marvin L. (1954). *Neural Nets and the Brain Model Problem*. Ph.D. thesis, Princeton Department of Mathematics.

Mollon, John (10 January 1985). "Colourful Notions: Studies in Scarlet". *The Listener*, pages 6–7.

Mowrer, Orval Hobart (1960). *Learning Theory and Behavior*. Wiley, New York.

Muggleton, Stephen (December 1995). "Inverse Entailment and Progol". *New Generation Computing*, **13**, pages 3–4.

Muller, Philippe (August 2002). "Topological Spatio-Temporal Reasoning and Representation". *Computational Intelligence*, **18**(3), pages 420–450.

Needham, Chris J., Santos, Paulo E., Magee, Derek R., Devin, Vincent, Hogg, David C. & Cohn, Anthony G. (September 2005). "Protocols from Perceptual Observations". *Artificial Intelligence*, **167**(1–2), pages 103–136.

Nisbett, Richard E. & Wilson, Timothy DeCamp (May 1977). "Telling More Than We Can Know: Verbal Reports on Mental Processes". *Psychological Review*, **84**(3), pages 231–259.

Pavlov, Ivan Petrovich (1927). *Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex*. Oxford University Press, Oxford, U.K.

Pearce, John M. & Hall, Geoffrey (November 1980). "A Model of Pavlovian Learning: Variations in the Effectiveness of Conditioned But Not of Unconditioned Stimuli". *Psychological Review*, **87**(6), pages 532–552.

Piciarelli, C., Micheloni, C. & Foresti, G. L. (November 2008). "Trajectory-based anomalous event detection". *IEEE Transactions on Circuits and Systems for Video Technology*, **18**(11), pages 1544–1554.

Pirri, Fiora & Reiter, Ray (May 1999). "Some Contributions to the Metatheory of the Situation Calculus". *Journal of the ACM*, **46**(3), pages 325–361.

Poesio, Massimo, Anderson, Andrew, Baroni, Marco, Eisenbeiss, Sonja, Rennie, Carol & Lenci, Alessandro (15 September 2011). "BabyExp: From Data Collection to Analysis". In Belz, Anja, Cosker, Darren, Keller, Frank & Makris, Dimitrios (editors), *Proceedings of the 2011 Vision and Language Workshop*, EPSRC Network on Vision and Language, Brighton, U.K.

Poesio, Massimo, Baroni, Marcoand, Lanz, Oswald, Alessandro, Lenci, Potamianos, Alexandros, Hinrich, Schütze, Schulte im Walde, Sabine & Surian, Luca (19–21 May 2010). "BabyExp: Constructing a Huge Multimodal Resource to Acquire Commonsense Knowledge Like Children Do". In Calzolari, Nicoletta, Choukri, Khalid, Maegaard, Bente, Mariani, Joseph, Odijk, Jan, Piperidis, Stelios, Rosner, Mike & Tapias, Daniel (editors), *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, pages 3042–3048. European Language Resources Association, Valletta, Malta.

Poole, David, Mackworth, Alan & Goebel, Randy (1998). *Computational Intelligence: A Logical Approach*. Oxford Universty Press.

Randell, David A., Cui, Zhan & Cohn, Anthony G. (October 1992). "A Spatial Logic based on Regions and Connection". In Nebel, Bernhard, Rich, Charles & Swartout, William (editors), *Proceedings of 3rd International Conference on Knowledge Representation and Reasoning*, pages 165–176. Morgan Kaufmann, Cambridge, Massachusetts, USA.

Rashotte, Michael E., Griffin, Robert W. & Sisk, Cheryl L. (March 1977). "Second-Order Conditioning of the Pigeon's Keypeck". *Animal Learning and Behavior*, **5**(1), pages 25–38.

Razran, G. H. S. (1939). "Studies in Configural Conditioning: I. Historical and Preliminary Experimentation". *The Journal of General Psychology*, **21**(2), pages 307–330.

Reiter, Ray (1991). "The Frame Problem in the Situation Calculus: A Simple Solution (Sometimes) and a Completeness Result for Goal Regression". In Lifschitz, Vladimir (editor), *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*, pages 359–380. Academic Press Professional, Inc., San Diego, California, USA.

Rescorla, Robert A. (August 1968). "Probability of Shock in the Presence and Absence of CS in Fear Conditioning". *Journal of Comparative and Physiologial Psychology*, **66**(1), pages 1–5.

Rescorla, Robert A. (April 1969). "Conditioned Inhibition of Fear Resulting from Negative CS-US Contingencies". *Journal of Comparative and Physiological Psychology*, **67**(4), pages 504–509.

Rescorla, Robert A. (May 1971). "Variation in the Effectiveness of Reinforcement and Nonreinforcement Following Prior Inhibitory Conditioning". *Learning and Motivation*, **2**(2), pages 113–123.

Rescorla, Robert A. (January 1973). "Effect of US Habituation Following Conditioning". *Journal of Comparative and Physiological Psychology*, **82**(1), pages 137–143.

Rescorla, Robert A. & Wagner, Allan R. (1972). "A Theory of Pavlovian conditioning: Variations in the Effectiveness of Reinforcement and Nonreinforcement". In Black, Abraham H. & Prokasy, William F. (editors), *Classical Conditioning II: Current Research and Theory*, pages 64–99. Appleton Century-Crofts, New York, New York, USA.

Robinson, J. A. (January 1965). "A Machine-Oriented Logic Based on the Resolution Principle". *Journal of the ACM*, **12**(1), pages 23–41.

Roy, Anthony J. & Stell, John G. (25–28 September 2002). "A Qualitative Account of Discrete Space". In Egenhofer, Max J. & Mark, David M. (editors), *Geographic Information Science, Second International Conference, GIScience 2002*, volume Lecture Notes in Computer Science 2478, pages 276–290. Springer, Boulder, Colorado, U.S.A.

Rummery, G. A. & Niranjan, M. (1994). "On-line Q-Learning using Connectionist Systems". Technical Report Technical Report CUED/F-INFENG/TR 166, Engineering Department, Cambridge University.

Russell, Stuart & Norvig, Peter (2003). *Artificial Intelligence: A Modern Approach*. Pearson Education, Upper Saddle River, New Jersey, U.S.A., 2nd edition.

Russell, Stuart J., Subramanian, Devika & Parr, Ronald (28 August – 3 September 1993). "Provably Bounded Optimal Agents". In Bajcsy, Ruzena (editor), *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 338–344. Morgan Kaufmann, Chambery, France.

Sadri, Fariba & Kowalski, Robert A. (June 1995). "Variants of the Event Calculus". In Sterling, Leon (editor), *Proceedings of the Twelfth International Conference on Logic Programming*, pages 67–81. MIT Press, Tokyo, Japan.

Samuel, Arthur L. (1959). "Some Studies in Machine Learning Using the Game of Checkers". *IBM Journal on Research and Development*, **3**, pages 211–229.

Santos, Paulo & Shanahan, Murray (July 2002). "Hypothesising Object Relations from Image Transitions". In van Harmelen, Frank (editor), *15th European Conference on Artificial Intelligence*, pages 292–296. IOS Press, Lyon, France.

Schmajuk, Nestor A. (2005). "Brain-Behaviour Relationships in Latent Inhibition: A Computational Model". *Neuroscience and Biobehavioral Reviews*, **29**(6), pages 1001–1020.

Schmajuk, Nestor A. (2008). "Classical conditioning". *Scholarpedia*, **3**(3), page 2316. Revision #73199.
URL: *www.scholarpedia.org/article/Classical_conditioning*
[accessed: 26 July 2013]

Schmajuk, Nestor A. & DiCarlo, James J. (April 1992). "Stimulus Configuration, Classical Conditioning and Hippocampal Function". *Psychological Review*, **99**(2), pages 268–305.

Schmajuk, Nestor A., Kutlu, Munir Gunes, Dunsmoor, J. & Larrauri, J. A. (2010). "Attention, Associations, and Configurations in Conditioning". In Schmajuk, Nestor A. (editor), *Computational Models of Conditioning*, chapter 6, pages 186–218. Cambridge University Press, New York, U.S.A.

Schmajuk, Nestor A., Lam, Ying-Wan & Gray, J. A. (July 1996). "Latent Inhibition: A Neural Network Approach". *Journal of Experimental Psychology: Animal Behavior Processes*, **22**(3), pages 321–349.

Schneiderman, Neil & Gormezano, Isidore (April 1964). "Conditioning of the Nictitating Membrane Response of the Rabbit as a Function of CS-US Interval". *Journal of Comparative and Physiological Psychology*, **57**(2), pages 188–195.

Schultz, Duane P. & Schultz, Sydney Ellen (1996). *A History of Modern Psychology*. Harcourt Brace College, Fort Worth, Texas, U.S.A, 6th edition.

Schweitzer, Laura & Green, Leonard (April 1982). "Reevaluation of Things Past: A Test of the "Retrospection Hypothesis" Using a CER Procedure with Rats". *Pavlovian Journal of Biological Science (now known as Integrative Physiological and Behavioral Science)*, **17**(2), pages 62–68.

Scivos, Alexander & Nebel, Bernhard (19–23 September 2001). "Double-Crossing: Decidability and Computational Complexity of a Qualitative Calculus for Navigation". In Motello, Daniel R. (editor), *Proceedings of the International Conference on Spatial Information Theory: Foundations of Geographic Information Science*, volume Lecture Notes In Computer Science 2205, pages 431–446. Springer, Morro Bay, California, U.S.A.

Shanahan, Murray (1999). "The Event Calculus Explained". In Wooldridge, J., Michael & Veloso, Manuela (editors), *Artificial Intelligence Today: Recent Trends and Developments*, volume 1600 of *Lecture Notes in Computer Science (LNCS)*, pages 409–430. Springer, Berlin, Germany.

Shortliffe, Edward H. & Buchanan, Bruce G. (April 1975). "A Model of Inexact Reasoning in Medicine". *Mathematical Biosciences*, **23**(3/4), pages 351–379.

Siegel, Shepard (May 1975). "Conditioning insulin effects". *Journal of Comparative and Physiological Psychology*, **89**(3), pages 189–199.

Singh, Push (March 2002). "The Public Acquisition of Commonsense Knowledge". In Karlgren, Jussi (editor), *Proceedings of AAAI Spring Symposium on Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*, pages 47 – 52. American Association for Artificial Intelligence, Palo Alto, California, U.S.A.

Singh, Push & Barry, Barbra (23–25 October 2003). "Collecting Commonsense Experiences". In *Proceedings of the Second International Conference on Knowledge Capture*, pages 154–161. ACM, Sanibel Island, Florida.

Singh, Push, Lin, Thomas, Mueller, Erik T., Lim, Grace, Perkins, Travell & Zhu, Wang Li (October 30 – November 1 2002). "Open Mind Common Sense: Knowledge Acquisition from the General Public". In Meersman, Robert & Tari, Zahir (editors), *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, volume Lecture Notes in Computer Science 2519, pages 1223–1237. Springer, Irvine, California, U.S.A.

Skinner, Burrhus Frederic (1938). *The Behavior of Organisms*. D. Appleton-Century Company, New York, New York, U.S.A.

Skinner, Burrhus Frederic (October 1962). "Two "Synthetic Social Relations"". *Journal of the Experimental Analysis of Behavior*, **5**(4), pages 531–533.

Smith, Marius C. (December 1968). "CS-US Interval and US Intensity In Classical Conditioning of the Rabbit's Nictitating Membrane Response". *Journal of Comparative and Physiological Psychology*, **66**(3), pages 679–687.

Sokolov, Eugene Nikolaievich (21–24 Feburary 1960). "Neuronal Models and The Orienting Reflex". In Brazier, Mary A. B. (editor), *The Central Nervous*

System and Behavior, Transactions of the Third Conference, pages 187–276. Josiah Macy Jr. Foundation, New York, U.S.A.

Solomon, Richard L. & Corbit, John D. (March 1974). "An Opponent-Process Theory of Motivation: I. Temporal Dynamics of Affect". *Psychological Review*, **81**(2), pages 119–145.

Speer, Robert (January 2007). "Open Mind Commons: An Inquisitive Approach to Learning Common Sense". In Havasi, Catherine & Lieberman, Henry (editors), *Workshop on Common Sense and Intelligent User Interfaces*, ACM, Honolulu, Hawaii, USA.

Speer, Robert, Krishnamurthy, Jayant, Havasi, Catherine, Smith, Dustin, Lieberman, Henry & Arnold, Kenneth (February 2009). "An Interface for Targeted Collection of Common Sense Knowledge Using a Mixture Model". In Conati, Cristina, Bauer, Mathias, Weld, Dan & Oliver, Nuria (editors), *Proceedings of the 14th International Conference on Intelligent User Interfaces*, pages 137–146. ACM, Sanibel Island, Florida, USA.

Sridhar, Muralikrishna, Cohn, Anthony G. & Hogg, David C. (21–25 July 2008). "Learning Functional Object-Categories from a Relational Spatio-Temporal Representation". In Ghallab, Malik, Spyropoulos, Constantine D., Fakotakis, Nikos & Avouris, Nikos (editors), *Proceedings of the 18th European Conference on Artificial Intelligence*, pages 606–610. IOS Press, Patras, Greece.

Stauffer, Chris & Grimson, W. Eric L. (23–25 June 1999). "Adaptive Background Mixture Models for Real-Time Tracking". In Draper, Bruce A. & Hanson, Allen (editors), *Proceedings of the 1999 Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2246–2252. IEEE, Fort Collins, CO , USA.

Stell, John G., del Mondo, Géraldine, Thibaud, Rémy & Claramunt, Chrstophe (12–16 September 2011). "Spatio-temporal Evolution as Bigraph Dynamics". In Egenhofer, Max J., Giudice, Nicholas A., Moratz, Reinhard & Worboys, Michael F. (editors), *Proceedings of the 10th International Conference on Spatial Information Theory*, volume Lecture Notes in Computer Science 6899, pages 148–167. Springer, Belfast, Maine, U.S.A.

Stell, John G. & West, Matthew (4–6 November 2004). "A Four-Dimensionalist Mereotopology". In Varzi, Achille C. & Vieu, Laure (editors), *Proceedings of the Third International Conference Formal Ontology in Information Systems*, pages 261–272. IOS Press, Torino, Italy.

Sutton, Richard S. (February 1984). *Temporal Credit Assignment in Reinforcement Learning*. Ph.D. thesis, Department of Computer Science, University of Massachusetts, Amherst, Massachusetts, U.S.A.

Sutton, Richard S. (August 1988). "Learning to Predict by the Methods of Temporal Differences". *Machine Learning*, **3**(1), pages 9–44.

Sutton, Richard S. (2010). "Errata and Notes for: Reinforcement Learning: An Introduction".
URL: *webdocs.cs.ualberta.ca/~sutton/book/errata.html*
[accessed: 26 July 2013]

Sutton, Richard S. & Barto, Andrew G. (March 1981). "Toward a Modern Theory of Adaptive Networks: Expectation and Prediction". *Psychological Review*, **88**(2), pages 135–170.

Sutton, Richard S. & Barto, Andrew G. (16–18 July 1987). "A Temporal-Difference Model of Classical Conditioning". In *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*, pages 355–378. The Cognitive Science Society, Lawrence Erlbaum Associates, Seattle, Washington, U.S.A.

Sutton, Richard S. & Barto, Andrew G. (1990). "Time Derivative Models of Pavlovian Reinforcement". In Gabriel, Michael & Moore, John (editors), *Learning and Computational Neuroscience: Foundations of Adaptive Networks*, chapter 12, pages 497–537. MIT Press, Cambridge, Massachusetts, U.S.A.

Sutton, Richard S. & Barto, Andrew G. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, Massachusetts, U.S.A.

Sutton, Richard S., Maei, Hamid Reza, Precup, Doina, Bhatnagar, Shalabh, Silver, David, Szepesvári, Csaba & Wiewiora, Eric (14–18 June 2009). "Fast Gradient-Descent Methods for Temporal-Difference Learning with Linear Function Approximation". In Bottou, Léon & Littman, Michael (editors), *Proceedings of the 26th International Conference on Machine Learning*, pages 993–1000. International Machine Learning Society, Montreal, Canada.

Taylor, Matthew E. & Stone, Peter (July 2009). "Transfer Learning for Reinforcement Learning Domains: A Survey". *Journal of Machine Learning Research*, **10**(1), pages 1633–1685.

Thompson, Richard F. (29 August 1986). "The Neurobiology of Learning and Memory". *Science*, **233**(4767), pages 941–947.

Thorndike, Edward Lee (June 1898). "Animal Intelligence: An Experimental Study of the Associative Processes in Animals". *Psychological Review: Monograph Supplements*, **2**(4), pages i–109.

Thrun, Sebastian (1992). "The Role of Exploration in Learning Control". In White, David. A. & Sofge, Donald. A. (editors), *Handbook for Intelligent Control: Neural, Fuzzy and Adaptive Approaches*, Van Nostrand Reinhold, New York, U.S.A.

Turing, Allan Mathison (October 1950). "Computing Machinery and Intelligence". *Mind*, **59**(236), pages 433–460.

Van de Weghe, Nico, Cohn, Anthony G., De Maeyer, Philippe & Witlox, Frank (November 2005). "Representing Moving Objects in Computer-Based Expert Systems: The Overtake Event Example". *Expert Systems with Applications*, **29**(4), pages 977–983.

Van de Weghe, Nico, Cohn, Anthony G., de Tre, Guy & da Maeyer, Philipe (2006). "A Qualitative Trajectory Calculus as a Basis for Representing Moving Objects in Geographical Information Systems". *Control and Cybernetics*, **35**(1), pages 97–119.

van Hasselt, Hado (2012). "Reinforcement Learning in Continuous State and Action Spaces". In van Otterlo, Martijn & Wiering, Marco (editors), *Reinforcement Learning: State of the Art*, pages 207–251. Springer.

van Rijsbergen, Cornelis Joost (1979). *Information Retrieval*. Butterworths, London, 2nd edition.

Vandercar, D. H. & Schneiderman, Neil (1967). "Interstimulus Interval Function in Different Response Systems During Classical Discrimination Conditioning of Rabbits". *Psychonomic Science*, **9**(1), pages 9–10.

von Ahn, Luis, Kedia, Mihir & Blum, Manuel (22–28 April 2006). "Verbosity: A Game for Collecting Common-Sense Facts". In Grinter, Rebecca, Rodden, Thomas, Aoki, Paul, Cutrell, Ed, Jeffries, Robin & Olson, Gary (editors), *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 75–78. ACM, Montréal, Canada.

Wagner, Allan R. (1981). "SOP: A Model of Automatic Memory Processing in Animal Behavior". In Spear, N. E. & Miller, R. R. (editors), *Information Processing in Animals: Memory Mechanisms*, chapter 1, pages 5–47. Lawrence Erlbaum Associates, Hillsdale, New Jersey, USA.

Wagner, Allan R. & Brandon, Susan E. (1989). "Evolution of a Structured Connectionist Model of Pavlovian Conditioning (AESOP)". In Klein, Stephen B. & Mowrer, Robert R. (editors), *Contemporary Learning Theories: Pavlovian Conditioning and the Status of Traditional Learning Theory*, pages 149–189. Erlbaum, Hillsdale, New Jersey, U.S.A.

Watkins, Christopher J. C. H. (May 1989). *Learning from Delayed Rewards*. Ph.D. thesis, King's College, Cambridge, U.K.

Watkins, Christopher J. C. H. & Dayan, Peter (May 1992). "Technical Note: Q-Learning". *Machine Learning*, **8**(3/4), pages 279–292.

Watson, John B. (March 1913). "Psychology as the Behaviorist Views It". *Psychological Review*, **20**(2), pages 158–177.

Witbrock, Michael, Matuszek, Cynthia, Brusseau, Antoine, Kahlert, Robert, Fraser, C. Bruce & Lenat, Douglas B. (March 2005). "Knowledge Begets Knowledge: Steps towards Assisted Knowledge Acquisition in Cyc". In Chklovski, Timothy, Domingos, Pedro, Lieberman, Henry, Mihalcea, Rada, & Singh, Push (editors), *Proceedings of the AAAI 2005 Spring Symposium on Knowledge Collection from Volunteer Contributors*, pages 99–105. AAAI, Palo Alto, California, U.S.A.

Wyatt, Danny, Philipose, Matthai & Choudhury, Tanzeem (9–13 July 2005). "Unsupervised Activity Recognition Using Automatically Mined Common Sense". In Veloso, Subbarao, Manuela; Kambhampati (editor), *Proceedings of the 20th National Conference on Artificial intelligence*, volume 1, pages 21–27. AAAI, Pittsburgh, Pennsylvania, U.S.A.

Xiaojin, Zhu (2008). "Semi-Supervised Learning Literature Survey". Technical Report 1530, Computer Sciences, University of Wisconsin-Madison.

Xu, Dong & Chang, Shih-Fu (June 2007). "Visual event recognition in news video using kernel methods with multi-level temporal alignment". In Baker, Simon, Matas, Jiri, Zabih, Ramin, Kanade, Takeo & Medioni, Gerard (editors), *Proceedings of the Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, IEEE, Minneapolis, Minnesota, USA.

Yilmaz, Alper, Javed, Omar & Shah, Mubarak (December 2006). "Object Tracking: A Survey". *ACM Computing Surveys*, **38**(4).

Yu, Victor L., Fagan, Lawrence M., Wraith, Sharon M., Clancey, William J., Scott, A. Carlisle, Hannigan, John, Blum, Robert L., Buchanan, Bruce G. & Cohen, Stanley G. (21 September 1979). "Antimicrobial Selection by a Computer: A Blinded Evaluation by Infectious Disease Experts". *Journal of the American Medical Association*, **242**(12), pages 1279–1282.

Zlatanova, Siyka, Rahman, Alias Abdul & Shi, Wenzhong (May 2004). "Topological Models and Frameworks for 3D Spatial Objects". *Computers and Geosciences*, **30**(4), pages 419–428.