FACULTY OF SCIENCE & TECHNOLOGY

BSc (Hons) Computing
May 2017

Four Laws and Neural Networks

by

Russell Allen Eaglesfield Clarke

Faculty of Science & Technology
Department of Computing and Informatics
Final Year Project

Abstract

Using an empirical approach, this project explores and implements the combination of immediate rewards with long-term rewards to study the behavioural effects on simulated environments using a selection of reinforcement learning (RL) solutions. The reward hypothesis (Silver 2015a) combined with immediate reward, emulates a form of human learning cognition where given a problem to solve, immediate reward is satisfied short-term and the experience is stored long-term for application in similar situations.

A new era of AI is upon us (Sedwill & Wallport 2016), the safety of humanities socio-economic state is paramount and there is an obligation to be better prepared for what may come. The concrete AI safety problems presented by Brockman and Christiano (2016), are core to operational capabilities of autonomous systems. There is a need for ethical and legislative standardisation of requirements and since there are no rules for how 'intelligent' autonomous machines should behave (Wakefield 2015), this project explores the behaviours and performance of RL to improve the avoidance of negative effects over the predecessors.

The modifications are implemented within a cross entropy method (CEM) and two different policy gradient (PG) solutions using various immediate reward values. A quantitative-qualitative approach yields 10 discreet simulations of 150 episodes for each implementation (1,500 episodes per solution). The results exhibit an average 33.7% increase in reward accumulation across all modified solutions, sufficient to minimise negative side effects with 5.4% improvement compared to original solutions.

From each modified PG and CEM solution an average 2.4% standard deviation is recorded. High value inhibitors (penalising for incorrect actions) exhibit promising results, outperforming other implementations by an average of 12.5%. These results satisfy the immediate objectives of this project and two of the three hypotheses partially.

The importance of investigating behaviours in artificial solutions and their responses to inhibitors and activators is critical if not obligatory to ensure the safety of society and its inhabitants. Standardising the engineering, legislative and safety frameworks will facilitate requirements and best practice to ensure the research of and transition into the 'new era' of autonomy, is approached with the cohesion and precautionary measures aiming to protect all involved. Without such research or frameworks, elements of risk are introduced into society, economy and industrial work places where autonomous systems are used.

Dissertation Declaration

I agree that, should the University wish to retain it for reference purposes, a copy of my dissertation may be held by Bournemouth University normally for a period of 3 academic years. I understand that once the retention period has expired my dissertation will be destroyed.

Confidentiality

I confirm that this dissertation does not contain information of a commercial or confidential nature or include personal information other than that which would normally be in the public domain unless the relevant permissions have been obtained. In particular, any information which identifies a particular individual's religious or political beliefs, information relating to their health, ethnicity, criminal history or sex life has been anonymised unless permission has been granted for its publication from the person to whom it relates.

Copyright

The copyright for this dissertation remains with me.

Requests for Information

I agree that this dissertation may be made available as the result of a request for information under the Freedom of Information Act.

Signed:  .

Name: Russell Allen Eaglesfield Clarke

Date: 7th May 2017

Programme: BSc (Hons) Computing

Original Work Declaration

This dissertation and the project that it is based on are my own work, except where stated, in accordance with University regulations.

Signed:  .

Acknowledgments

I would first like to thank all those whom have got me this far, not just in my studies but in life. In particular, I would like to thank the faculty of Bournemouth University, without their support and wisdom over the last three years, this dissertation and accompanied project would not have been possible. To my Partner Holly for proof reading this work and putting up with the long hours and supporting me every step of the way.

A special thanks and recognition to my Project Supervisor, Professor. Marcin Budka. Without his support, critique and positive attitude in those bleak ‘I’m so worried …’ moments, this would never have been possible. To Sue Churchill for her assistance in developing my written skills over the last three years.

And so, to all of you, you know who you are, I am sincerely grateful for all you have contributed to make this happen.

Thank you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 BACKGROUND AND CONTEXT

Within the category of Artificial Intelligence (AI) there are many disciplines, Machine Learning (ML) is one such discipline which hold a set of problem spaces and ethical dilemmas. Challenges facing progress in AI and learning systems are numerous, particularly where benchmarks are insufficient (Brochman & Schulman 2016).

Additionally, there appears to be general confusion about AI and its related disciplines, where assumptions may be made that all AI systems are similar although they are not. AI is a category of research and engineering with sub-categorical disciplines and approaches. ML and Robotics among others, each hold their own set of problem spaces and differ in application and approach.

One branch of ML is Reinforcement Learning (RL); a branch of ML aiming to enable safe, autonomous learning across multiple application domains.

> *"We don't currently have any rules for how robots should behave if and when they start operating autonomously" (Wakefield 2015).*

A scalable and general purpose ML solution requires an exploratory approach in its design and research to satisfy operational requirements for application in autonomy. Furthermore, the RL problem is potentially one of the most difficult to solve given the erroneous behaviours exhibited by RL agents (Clark 2016), multiple tasks in dynamic environments and in consideration of ethics, safety and human values.

1.2 PROPOSED SOLUTION

It is important to understand an ‘intelligent’ machine cannot simply be set into society without sufficient ability to refrain from bringing harm to its inhabitants. One of the largest problems in AI is the manner in which an agent learns. The empirical approach, based on observation and experience, applied to a machine without sufficient training prior to ‘active duty’, could be dangerous as the agent or machine use a trial and error approach in establishing boundaries of their environment and applicable actions. Thus, in RL, the potential solution space for learning would be to learn correct actions quickly, either by being trained with data (audible, visual, kinetic, syntactic) or learning fast enough to avoid negative impact but preferably, a combination of both.

1.3 AIMS AND OBJECTIVES

This project has immediate objectives to perform exploratory research which yields quantifiable results to compare performance of current RL solutions against this projects’ RL implementations. This project defines criteria of success to be measured on ability to learn by minimising negative behaviour. However, the broad criteria of the RL problem are to have no negative behaviour and safe operation within societies diverse environmental dynamics.

1.3.1 Aim

Taking an exploratory approach of investigation into the behavioural properties of ML, exploring RL and how the performance is enhanced or inhibited with various modifications to the implementations and reward values.

The immediate intended aim, derive sufficient results to confirm the possibility of basic behavioural intelligence and how this might be achieved in computing systems by comparing the effects to different reward values in RL solutions.

1.3.2 Objectives

Source, examine and implement several known RL algorithms which may be applicable in extrapolating sufficient results to determine the effects of Reward Value Modification (RVM) introduced to effect 'intended' behaviour of the original solution.

Collect and analyse sufficient data to represent the effects and trends RVM implementations have had on Agent's performance, in comparison to the original implementation and to each other.

1.4 Definition of the problem

RL is based on the reward hypothesis "*All goals can be described by the maximisation of expected cumulative reward*" (Silver 2015a); therein, resides the ML problem, being able to develop a robust agent which reacts predictably and safely in an environment to accomplish tasks autonomously.

While supervised and unsupervised methods can attain a specific goal from untrained or trained standpoint respectively, these types of agents are trained or interacted with for one specific task or several. RL however, must learn from its own experience and a potentially delayed reward signal, with its suitability for production in machines being measured on its performance at a range of tasks within dynamic environments.

The agent is the implementation of interest. It receives an observation, a stream of data related to current state, performs actions and later receives a reward signal accompanied with observation of state. The values received by the observation and reward, represent whether previous actions taken were good or not. An agent' capability should satisfy concrete AI safety problems (Brockman & Christiano 2016) and meet a minimum set of requirements:

- *Safe exploration*
- *Robustness in distribution shift*
- *Avoiding negative side effects*
- *Avoiding reward hacking or wire-heading*
- *Scalable oversight*

The agent should be scalable meaning, it should be able to perform in a robust, predictable and satisfactory manner in both large scale (online) and in autonomous deployments (robotics). It also needs to perform safe operations in keeping with human legislation, ethics and values (UK Government 2016; European Parliament 2016; IEEE 2016).

As such, the RL problem is not just how the agent learns but also how capable it is in satisfying all requirements and expectations long-term meaning, there is finiteness to some new 'solutions'. Other derivative requirements may be task specific, building an automobile or caring for an elderly human for example, are tasks with unique subsets of requirements.

1.5 PROJECT OVERVIEW

Though desirable, it will not be possible to cover every discipline in-depth, as such there is considerably more research and investigation required hereafter. The following chapters provide an insight to Neural-nets (NN), OpenAI, AI, ML and Ethics. Analysis of the requirements for a suitable agent is provided and designs are synthesised. Implementation and results are evaluated and further discussion of future work is held in conclusion of successes or failures suggesting other potential solutions.

1.6 SUCCESS CRITERIA AND RISK MANAGEMENT

1.6.1 Success criteria

Current known solutions are used initially to gain an understanding of RL implementations and secondly to gather results. Modifications will be made to attempt to improve the current implementation. It is important to note the RL problem cannot be solved within the duration of this project. Success will be measured on improving current known implementations such that the time it takes an implementation to solve a given problem, is less than its predecessor or that distinct differences in performance are observed.

1.6.2 Risk management

Due to time constraints, the risk to project failure is managed using Gantt charts where the objectives and aims are monitored, tracked and adapted (where necessary) to ensure the scope is maintained and the project kept within limitations. Additional constraints are that of implementation and limited experience with Python programming language mitigated with practice, previously acquired programming concepts and online resources Python (2017a) twinned with the toolkit Gym, hosted by OpenAI.

Risk Management strategy			
Description	*Impact (10 = highest)*	*Likelihood (10 = highest)*	*Mitigation*
Time	10	9	Chronological, functional and logical planning. Monitoring progress with Gantt chart.
Ability	9	7	Research, practice, understand, focus. Meetings with supervisor.
Ethics	7	6	Checklist, conformity and review.
Environmental	9	3	Preventative measures and back-ups of work.
Sickness	7	4	Health, hygiene and doctors.
Personal Life - Children, appointments and Home duties.	6	5	Time management and separate study times and space.

Table 1 Shows current risk management strategy (Author 2017).

2 BACKGROUND STUDY

2.1 OVERVIEW

This chapter covers most relevant background information of topics discussed in this dissertation aiming to define characteristics, variants, applicable technique and summarise with any current 'real-world' applications their advantages, disadvantages or future potential if any.

2.2 ARTIFICIAL NEURAL NETWORKS

The formal definition of a NN given by Hecht-Nielsen and Nielson (1990 p.2-3) in summary is; A parallel, unidirectional, distributed processing system for which the localised processors storage relies on its own weighted restrictions and the multiple input it receives to generate a distributed identical output. Essentially what this means is an array of input is received and based on that information and the processors function, it generates and distributes an output. This relates to the classic architecture of input-process-output model in Computing Systems (see Figure 1) described by Hecht-Nielsen and Nielson (1990, p.23) and Englander (2003, p.9-11).

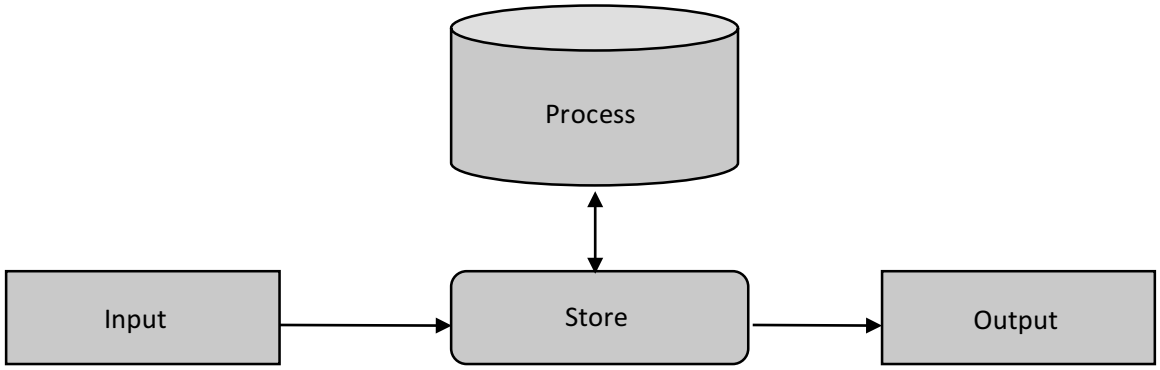


Figure 1 Input-Process-Output model (Adapted from Englander 2003, p.10).

2.2.1 Characteristics of Neural-Nets

An NN could be akin to a human brain given more complexity; however, this notion should be refuted as NN share no resemblance to the brain, in part due to its complexity and centred on the notion we still know very little about its function after more than a century of research according to Hecht-Nielsen and Neilson (1990, p.12-13) and Müller et al. (1995 cited in Nauck et al. 1997 p.11-12).

The nearest resemblance is the conceptual representation of the NN 'node' to a neuron (see Figure 2). This is not to say there is no relationship between *neuroscience* and *neuro-computing*, the two disciplines have begun to exchange ideas and results to understand the human brain and improve research (Hecht-Nielsen & Neilson 1990, p.12-13; Cox & Dean 2014, p.922). Thus, the only reference to biological neurology hereafter, shall be explicitly stated and further discussion on NN refers to the artificial.

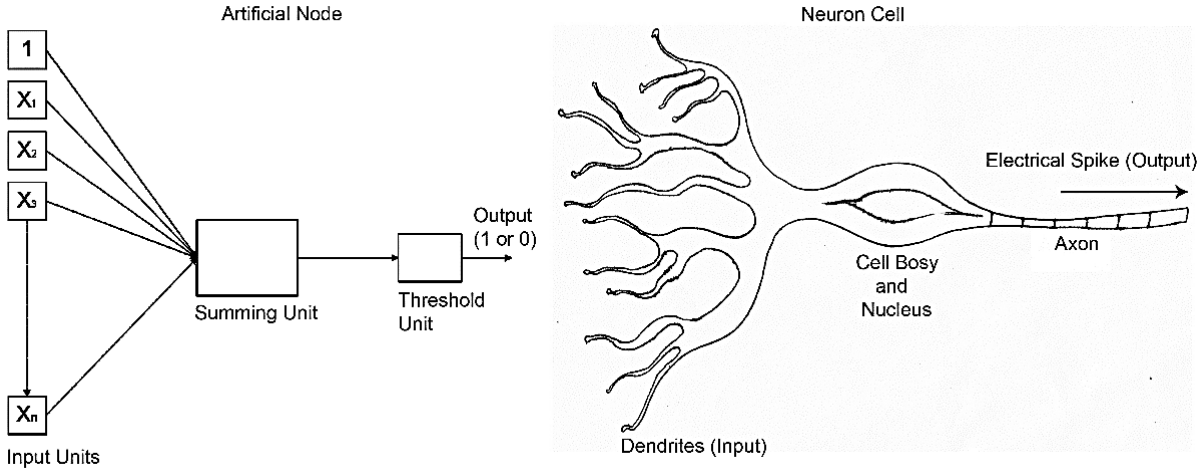


Figure 2 Comparison of Artificial Node to Neuron (Adapted from Whitby 2008, p.46).

Although many scholars in the field of neuro-computing (Hecht-Nielsen & Nielson 1990, p.22-23; Nauck et al. 1997, p.10-11; Whitby 2008, p.46) tend to disagree over the standardisation of a node, the concept is effectively the same;

- There are multiple inputs to each node from multiple sources.
- Each processing unit or 'node' is either activated or inhibited by an input, threshold or time-scheduling, it has local storage which may or may not be modified based on the nodes local ruleset and has a fixed threshold for generating an output. A node can receive any number of inputs but produces a single output.
- Each output from a single node is identical even when replicated until the inputs of the node cause a change to the processed output. Each output either becomes an input to another node, itself (recurrence), or leaves the network.
- The direction of data flow is unidirectional.

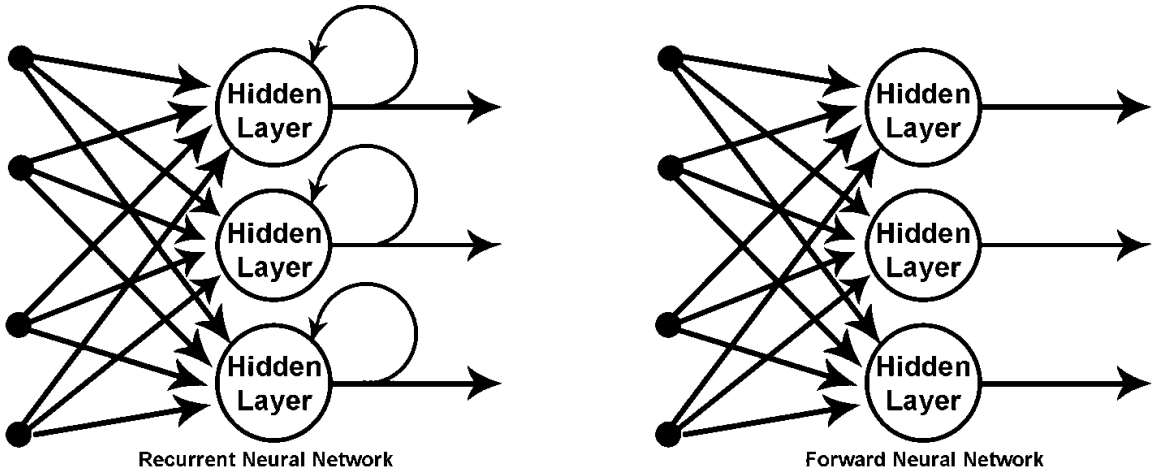


Figure 3 Comparison of RNN and FNN (Amended from Bethard et al. 2014).

2.2.2 Main Neural-Net structures

Aside from basic 'feed-forward' NNs (FNN), unidirectional flow is not strictly true as a characteristic; Continuous Time Recurrent NNs (CTRNN) and Recurrent NNs (RNN), allow for recurrent connections of output signals to their own node. The purpose of an RNN or CTRNN is to

reproduce complex dynamics emulating temporal sequence learning utilising algorithm as functions. The time factor in CTRNN allow for a consideration of time dynamics in the networks learning function (Arbib 2003, p.15; Tani & Yamashita 2008, p.3). Thus, one distinct advantage of CTRNN will be to learn from previous state and a key advantage to RNNs, their ability to preserve local state (memory).

In describing dynamic systems, Arbib (2003 p.15) separates the processing node into *state* and *state transition function*, where the state (set(s) of variables) is determined by the node's environment (input(s) and/or output(s)) and changes per its function. Tani and Yamashita (2008), favoured CTRNN on its ability to maintain state, in doing so have omitted the desired function to learn or adapt to a changing environment which suited their experiment however, dynamic CTRNN may prove useful in other experiments where the physical environment is not immediately known and changes (moving target).

A dynamic CTRNN could be considered an adaptive system. The process involves an interaction with an environment and selecting an appropriate previously learned model of the problem, adapting to new environment variables to provide a solution (Arbib 2003, p.18). Although CTRNN have exhaustive runtime, one benefit to adaptation is modifying previous states to better recognise and solve problems.

Contrary to FNN, bidirectional networks (BRNN) store and split vector pairs in a bipolar manner. On output, the weighted signal is bifurcated as dual-opposing output to aide optimisation via associated memory and pattern processing in NN layers (Sun 2017). Furthermore, Sun (2017) proceeds to explain BRNN synonymy with RNNs and dynamic systems insisting CTRNN be evaluated in discrete counterparts, allowing for quantifiable results during computation.

Discrete simulation counterparts may be required during this project to avoid exhaustive runtime. Several drawbacks to BRNN depending on use-case, are extensive training time, complexity and training data requirements (Bethard et al. 2014, p.65; Arisoy et al. 2015, p.21). Other NN types include Cascading among others, these types of neural-traversal and complexities, are beyond the scope of this dissertation.

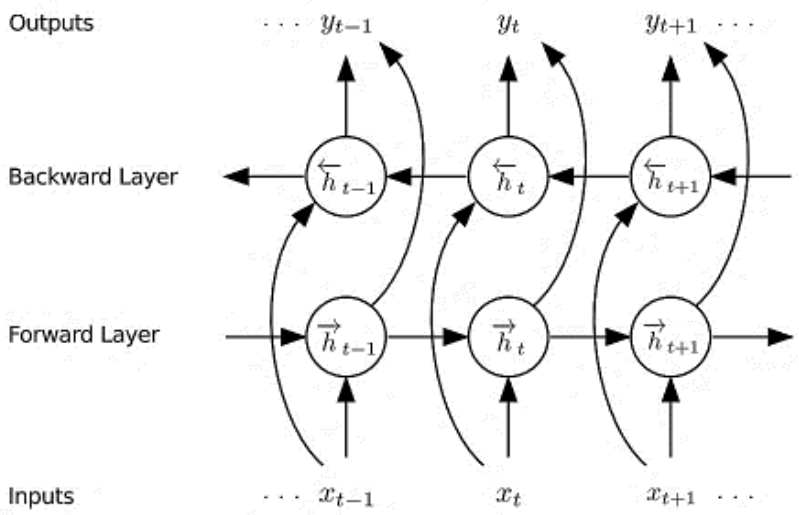


Figure 4 Bidirectional RNN showing bifurcated output concept (Graves et al. 2013).

2.2.3 Neural-Net implementation and considerations

Typically, NNs are implemented with procedural, functional or object orientated programming and make use of calculus in NN implementations to automate NN weight adjustment. Examples can be found by Miller (2015) and Karpathy (2016b). Solutions by Miller and Karpathy(2016a), utilise automation to update the NN weights, this enhances association between nodes and improves synaptic connection distribution across the network (Nauck et al. 1997, p.2).

Kahn (cited in Jain et al. 1999, p.110) highlights difficulties in determining initial NN size to solve a specific problem. With a lack of potential for scalability, NNs may not prove to be cost effective during implementation, still an arguable topic given recent developments (de Campos et al. 2017) since Keuper and Preundt (2016).

Examples presented by Miller and Karpathy require substantial run-time for the RL agent to learn how to perform optimally (where the agent is considered the 'brain'). When running a solution by Karpathy, it took 3-4 days before seeing any substantial results and Miller also highlights long training times in NNs.

Furthermore, frameworks or toolkits are also in use for the evaluation and testing of RL implementations. One such toolkit is OpenAI Gym (2016c). OpenAI (2016a) is a non-profit, RL research company and Gym is their toolkit for testing potential RL solutions, discussed in subsequent chapters. Exploring the RL problem using Gym toolkit (OpenAI 2016c), current known solutions will be examined for suitability and after running the implementations, the results will be collected for comparison.

In addition, physical NN systems are also in development described as Memristor by Min et al. (2017) and Shin et al. (2017); Memristor systems (Palmer 2012) favour a processor implementation over the diodes, resistors and circuits originally used in the ADALINE systems developed by Widrow (1963).

2.3 Toolkit and Implementation

Several benefits arise from a toolkit pertaining to quantifiable and comparable results from a variety of implementations. Environments emulated by the Gym toolkit, are equal throughout all evaluations despite the developers own implementation method. The toolkit provides a unified general testbed suitable for comparison and reproduction of RL solutions and can allow for priority focus on the implementation of the RL solution as opposed to the backend system development.

2.3.1 OpenAI Gym

OpenAI (2016a), is a not-for-profit AI research company with a primary focus and goal to evaluate and validate safe AI soft solutions for the benefit of humanity. The very core of OpenAI is community led creation of general purpose algorithms which perform optimally at a range of tasks and do so safely (Altman et al. 2016). In doing so, OpenAI have released a toolkit (Gym) as a testbed, hosting multiple emulated environments since 2016. Since Gym, another toolkit named Universe has also been released with a plethora of higher complexity environments to include but not limited to, emulated webpages, comparing and validating 'general purpose' algorithm suitability across vast environments and task variance.

2.3.2 OpenAI Gym, API and development languages

The Gym toolkit is currently implemented using Python (2017a) programming language (though other language support is anticipated) and supports a variety of mathematical implementation frameworks (Brockman & Schulman, 2016).

2.3.3 Current implementations, advantages and limitations

Nielsen (2015) has reported potentially converting a handwritten character recognition NN to Javascript for web deployment which could be beneficial to companies or individuals wanting to search digitised handwritten documents.

One distinct advantage in being able to potentially port RL solutions as server scripts, is the sheer compute power available via cloud technologies post 2016 since client-side scripts (Nielsen 2015) may consume too much compute resources on the user's machine. Cloud technology would permit for higher performance and faster training times and include scalable resources for large projects. In addition, capacity planning may not be required for smaller projects however, it should be noted with higher complexity NNs and larger training datasets, comes longer training times despite compute power and capacity (Nielsen, 2015).

Moreover, as NN and AI technology advances, serious ethical consideration to the suitability of using the Internet for deployment or as a testbed should be discussed; particularly where the full outcome or operable capability of the implementation is not known and owing to the fact most strengths and weaknesses of humanity are hosted online as an information infrastructure we heavily rely on (Cerf, 2013).

2.4 Artificial Intelligence

It may be difficult to define intelligence due to a lack of comparable benchmarks. Since there is no knowledge of anything exceeding human level, there is nothing to compare the human upper limit to, nor to definitively place the human intellect hierarchically. However, defining a state of intelligence could be as an ability to learn, retain and apply information to provide solutions to problems. In addition, human level intelligence could be used in defining a suitable benchmark for achieving AI within the realms of human socio-economic state.

> *"AI embodies an inexact form of computation which appears, at best, to be based on an ambiguous model (similar to human reasoning)"*
> (Schalkoff 1990, p.xxii).

2.4.1 What is Artificial Intelligence?

It is generally agreed, AI encompasses many disciplines and applications (see Figure 5). However, what may be AI one day becomes obsolete or benign on another; primarily due to change as specific AI solutions are scrutinised for industrial suitability (Tsujii & Shirai 1989, p.1; Schalkoff 1990, p.1-4; Warwick 1991; Silver 2015a).

It becomes difficult to define AI and though 'artificial' is simple to define, intelligence is not. Intelligence however, is understood to be combinations of logical, lateral and creative cognition applied in establishing and solving problems within a dynamic environment (Gunderson & Gunderson 2017, p.1).

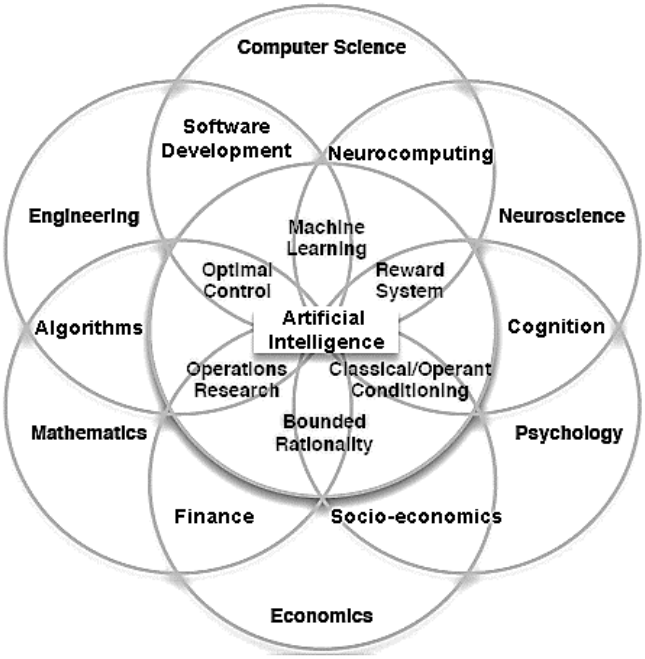


Figure 5 AI applications and disciplines (Amended from Silver 2015a).

The discipline of AI, may be defined as studies into the possibility of machine intelligence and automation capable of human level problem solving which may be classed into two approaches; the Engineering approach, concerned with application of methods and the Exploratory approach, concerned with empirical research in establishing methods (Schalkoff 1990; Cohen 1995, p.2-9).

2.4.2 AI implementation

While AI may be considered as autonomy, the two are generally classified separately. Consider the engineering approach, Schalkoff (1990, p.2-3) elaborates on this as a solution brought about by representative autonomy for human provided solutions. This Implies, a product doing something automatically which humans know how to do already, agreed by Gunderson and Gunderson (2017 p.2).

However, Gunderson and Gunderson (2017) argue AI and autonomy are separate. Autonomy in effect, is repetitive and far from sporadic or adaptable; synonymous to the engineering approach, AI is not just automation of a simple process but setting goals for an agent to reach autonomously argue Sedwill and Walport (2016, p.1).

Introspectively, although there are many disciplines in AI and factoring in potential to eventually have AI applied in humanoid robotics to aide in human socio-economic situations, it is difficult to say autonomy is classified separately of AI, particularly with ‘automatic’ being at the core, a fundamental requirement. To elaborate, without autonomy artificial agents cannot solve a problem in a dynamic environment thus, the agent is not suitable. In this, it could be said a far-sighted goal has not yet been sufficiently reached. However, AI has been implemented in engineering industrial and social solutions, examples include automotive, telecommunications and circuit manufacturing to name a few.

2.4.3 Current limitations and advantages of existing AI systems

In addition to Industrial applications, financial, linguistic and retail applications of AI are becoming widespread throughout (Sedwill & Walport, 2016).

"artificial intelligence is enabling a new wave of innovation across every sector of the UK economy" (Sedwill & Walport 2016).

With prediction and diagnostic abilities through to pattern recognition and data analysis, the autonomous nature of AI is having a profound impact on economic growth though with limited progress due to the capabilities and performance of technological improvements and an absence of definitive benchmarks say Sedwill and Walport (2016 p.6-7). Discussing the move from AI to ML, Sedwill and Walport highlight two common ML types, unsupervised and supervised.

2.5 Artificial Learning

Supervised learning is an interactive approach to teaching an agent to perform tasks with the use of training data. In contrast, unsupervised learning is the definition of a behavioural method (algorithm(s)) to establish data patterns (Sedwill & Walport 2016, p.6). The ethical and safety issues for some types of unsupervised learning are of concern, particularly where the agent may not 'know' how to achieve the goal and takes an empirical approach in establishing the best method to do so. This in turn may have many potentially negative or hazardous implications.

2.5.1 AL differs from AI

Considering AI is a classification to an area of research which encompasses many disciplines, Artificial Learning (AL) is one such discipline with several fields of expertise thus, AI and AL differ in their classification as categorical and specialism respectively. In addition to unsupervised and supervised learning techniques, RL has emerged as a promising learning technique, focusing on simulating human learning behaviour patterns by mapping observations to actions in receipt of numerical reward signal (Barto & Sutton, 1999).

2.5.2 Reinforcement Learning and Ethics

The agents' goal/policy provides a required accumulation of reward based on actions mapped from within the environment; using the action-map policy and resulting value function, agents learn to enhance performance for higher reward based on observation (Barto & Sutton, 1999; Silver 2015a). The mechanism of reward is either immediate or long-term (predicted). Though immediate is preferred by animals and humans, delayed reward signals are used in RL to maximise long-term reward and improve learning capabilities (Silver, 2015a). Barto, Sutton and Silver, among others, recognise a division of the RL solution into four broad components:

- Policy – Mapping state to actions and vice versa
- Observation – An internal representation of the environment (State)
- Reward Function – A reward function providing evaluation of environment interaction
- Value Function – Evaluated success of policy state rewards for long-term evaluation

The division also permits a base-structure for implementation as observed in potential solutions from Miller (2015), Karpathy (2016a) and toolkits provided by OpenAI. Contemplation on the concept of RL places it between supervised and unsupervised learning. To elaborate, the agent knows nothing of its environment only of its state, an internal ambiguous representation of the environment and its action validity (see Figure 6). The learning method can be recognised as the Markov property where the past is discarded as irrelevant given the current state and all potential future states, explains Silver (2015a; 2015b). The combination of an agent not knowing its environment explicitly and uncertainty of how to acquire reward before accumulation twinned with a

goal of maximising rewards as a learning strategy, places RL as semi-supervised learning though similar to unsupervised.

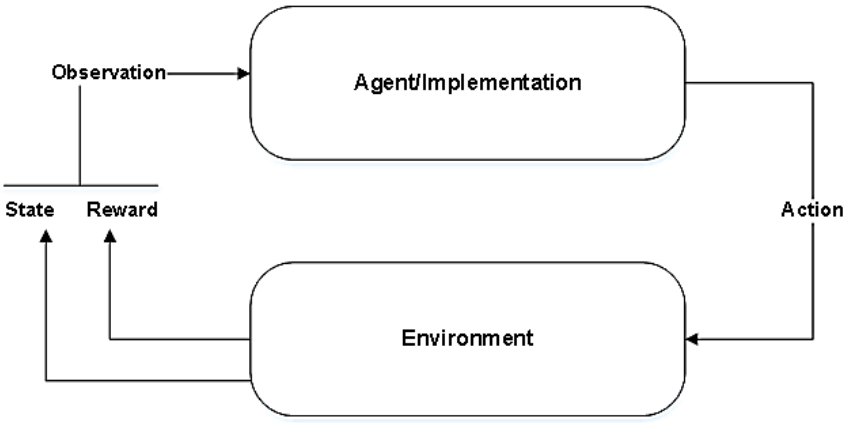


Figure 6 State transition model for RL, showing at each time step an agent takes an action based on an observation of state and reward from the environment (Adapted from Abbeel & Schulman 2016; Silver 2015a).

In their current state of validity, learning methods applied to robotics are not currently deemed safe as such, experiments such as those exhibited by Tani and Yamashita (2008), Miller (2015) and Karpathy (2016b) are performed in controlled environments and will probably continue to be controlled for the foreseeable future.

2.5.3 Ethical considerations

Recognising the ethical and safety considerations of AI at an early stage in its practice and development, could arguably be one of the most important disciplines in AI research. The UK Government (2015 and 2016) have brought into context, frameworks applicable to legislation which should be considered during all stages of AI development, clearly and obviously to ensure security and safety of humanity and its socio-economic state. The European Parliament (2016) have also released a pre-emptive draft framework explicitly for AI and Robotics and the IEEE (2016) are in the process of standardisation discussions with serious debates surrounding ethics and autonomy scenarios.

With hysteria and fears of AI in society (Wakefield 2015) the reality is AI and autonomy already exist. Although in its infancy, a prime example of negative bi-product from RL implementation is reward hacking; the agent learns it can repetitively accumulate large amounts of rewards by looping through an action despite the overall end goal (Clark 2016). In a hypothetical real-world scenario, an agent may learn, by committing robbery, it will secure enough resources to upgrade its own hardware and perform tasks more efficiently thus, gain more reward long-term. OpenAI (Brockman & Christiano 2016) address some of the concrete AI safety problems which include;

- **Safe exploration:** Learning about the environment without causing issues
- **Robustness in distribution shift:** Adaptability for changing data and safe failover
- **Avoiding negative side effects:** Modifying the reward function without negative impact
- **Avoiding reward hacking or wire-heading:** Creating more mess to satisfy cleaning rewards
- **Scalable oversight:** With minimal interaction from a human, can a robot still perform specified task to satisfaction

Factoring in dynamic environments, it would be unethical to release a humanoid product into society without sufficient training about its environment and interactions, it may have to destroy several buildings or damage several cars before learning to simply walk around them. RL implementations still need to be trained in their environment to learn how to perform correct actions and gain rewards however, this is partially supervised for precaution.

These considerations are typical examples to be aware of when discussing the ethics, operations and capability of AI agents and machines. Ethical considerations and standards can be used as a framework, ensuring engineering and research is performed for the right reasons with an end goal to engineer solutions which both transition into and help humanity positively and safely in keeping with human values and legislation.

2.6 SUMMARY

While NNs can have complexity, provide weighted outputs suitable for statistical analysis to establish patterns, their importance as a bottom-up approach to the field of AI may be a key element in securing the far-sighted goal of agents in society. However, one concern is their presence online and in large complex implementations, it may be difficult to determine how the NN arrived at its conclusion.

The field of AI contains many disciplines and from the ML standpoint, three distinct methods have been established; where supervised systems are interactively trained and the unsupervised are given a goal to achieve, RL lies somewhere in between as a semi-supervised method for ML. In contrast to bottom-up, the top-down approach explained by Bochman (Rossi 2013, p.782), aims to establish a general-purpose solution which covers multiple problem spaces and performs in a safe optimal manner however.

There are many ethical deliberations with mixed opinions and ambiguous benchmarks surrounding NNs, RL and in its entirety, AI. Among others, OpenAI maintain an ethical ethos, promoting and evaluating safe general-purpose algorithms which will eventually benefit humanity. Their Gym toolkit allows for dynamic implementation of NNs and RL with an aim to progress AI and validate potential application and safety in the respective field.

Arguably, as it has not yet been achieved sufficiently, establishing a general-purpose scalable solution which defines behaviour in autonomous ‘intelligent’ systems in anticipation for safe integration into society, may be one definitive benchmark closest to current AI research, albeit furthest away from the far-sighted. As the specification requirements for a scalable, safe solution are not immediately known nor until the benchmark is reached, other questions to the possibility of human level intelligence in machines or whether machines will supersede humans cannot be satisfied.

3 REQUIREMENTS AND ANALYSIS

3.1 OVERVIEW

It is essential to establish requirements and benchmarks relevant to aims and objectives, which can be worked through methodically.

3.2 PROBLEM STATEMENT IN CLARIFICATION

It is important to explore and investigate the potential of a general purpose, scalable RL solution which defines appropriate behaviour in autonomous 'intelligent' systems in anticipation for safe integration into society. It is also important to compare the suitability of current known RL solutions and make improvements to them if possible. Since there are limitations, the scope of this project explores effects of immediate reward signal on agent behaviour as its primary focus to satisfy the avoidance of negative side effects requirement.

3.3 HYPOTHESES

i. *Logical combinations of RVM may positively affect the agent' actions on the environment, if immediate rewards were gained for actions ahead of final reward value.*

ii. *Immediate rewards may not prove to possess efficiency, scalability, safety, ethical or 'general-purpose' requirements for introduction into society.*

iii. *Combinations of high negative or high positive reward values may not be suited to specific algorithms nor prove effective for all tasks.*

3.4 METHODOLOGY

3.4.1 Overview

Exploring the effects of RVMs on artificial intelligent agents and the performance or interactions within their environments, requires consideration of the methodological processes required.

3.4.2 Methodology considerations

Several methodologies and processes have been considered for this research. An overview can be found in Table 2.

Methodology / Process	**Type**
Empirical	Investigative, Exploratory
Qualitative	Generative, Reactive
Quantitative	Cumulative, Restrictive
Incremental	Replicative, Finite
Evolutionary	Iterative, Infinite (potentially)
Hybrid	Associative, Adaptive

Table 2 Overview of considered methodologies and processes (Author 2017).

The characteristics, methods and brief justifications of the methodologies and processes outlined in Table 3, can be found in Appendix D.2.4 with higher levels of detail (Adapted from Bausell 1986; Mason 1996; Le Blanc & Stiller 2002, p.43-48; Hughes & Ilbury 2014).

Quantitative research could be a suitable methodological approach to controlling collection of data and aide the verification or validity of a hypothesis and research strategy. A qualitative methodology, may be best suited for designing a suitable method to conduct research and reacting to unexpected anomalies. In addition, sorting, organising and indexing qualitative data or generated ideas can prove to be cumbersome without methods described by Mason (1996, Ch.6). Considering such methods, ensures an appropriate strategy of generation, collection and control for large quantities of primary and secondary resources.

Furthermore, an important factor to integrating quantitative and qualitative methodologies is that they could both be used interchangeably to ensure a robust set of methods during an empirical approach throughout the practical, generative and conclusive phases of research.

Since empirical research can seem awkward during exploration, its practical advantages potentially yield unexpected yet valid results. A key advantage to empiricism resides in generating quantifiable representations for observed phenomena (Bausell 1986, p.3-18), well-suited to exploratory research. In addition, after reviewing the empirical methodology (Bausell 1986), it is evident empiricism may enforce or even compliment methods and strategies used in qualitative or quantitative methodologies and potentially, incremental or evolutionary design processes where testing, validation or verification stages are concerned.

Incremental design processes, possessing finite development life-cycles and throwaway prototypes (Le Blanc & Stiller 2002 p.43-48), demonstrate suitability for exploring RVM implementation and once this has been achieved, redundant prototypes could be disposed of. Additionally, specific phases (Maintenance, Design) may be scaled down or even omitted during exploration or implementation however, not without foresight.

To compliment an incremental development life-cycle, an evolutionary process which, at each iteration is composed of some incremental phases (Le Blanc & Stiller 2002, p.46-48); can be used to facilitate long-term aims for scalability once a suitable agent has been established, possessing key benefits to functional prototypes and verification. Furthermore, the methodologies could be used throughout each iteration enabling a robust continual evaluation of the agent and potentially provide statistics on future milestones and objectives or predictive statistics relevant to distant benchmarks.

Excluding an evolutionary life-cycle (as it is not immediately required during this project), a hybrid approach will be used integrating the methodologies and processes discussed, holding potential to encapsulate their respective methods suitable for specific tasks. The combination of several methodological approaches and methods, compliment various aspects of this research including the design and production of an artefact.

3.4.3 Methods from Methodology

A hybrid methodology and its encompassing methods, could translate to methods applicable for current or upcoming tasks during the project. If anomalies are noticed in the qualitative result collection stages for example, the problem could be researched quickly for a root cause or solution

and remedied or modified in a reactive manner while still having the controlled collection methods permitted from quantitative methodology. Furthermore, generating large data sets of agent behaviour is anticipated for this research thus, qualitative and empirical methods will be best suited for analysis and representation of this result data under the supervision of quantitative verification and control.

3.4.4 Data collection

Deciding which data is relevant for analysis primarily depends on the size, purpose and output of an agent. Since the hypotheses generate specific requirements, quantifiable data should be collected for measuring and comparing the success of specific agents and RVMs:

- Time will be an important factor in comparing efficiency, learning ability and duration.
- Rewards gained during an episode or game, measure an agent' success for reward accumulation. In addition, a running mean of rewards remaining (Karpathy 2016a), appears to be useful for comparing learning performance.
- Collected across 10 simulations yielding 150 episodes in each, discreet sets of results will be collected for quantification and comparison. The RVMs are based on rewards where they are not issued, high value rewards for positive actions and high inhibiting values (punishing) for negative behaviour. An episode is a portion of time dedicated to solving a given problem and they can end in any manner of ways, a round of games in Pong with a winner or even an attempt at beating a timer for example (see Table 3).
- Video recordings of some trials may be useful to document observations of performance and behaviour and would serve as a tool to accompany written and numerical representation.

Developer	Agent No. / Developer Solution	Trials / Agent No.	Episode count (sample) / Trial
Kevin Frans (CartPole)	1 (Original solution, no RVMs)	10	First 150
Andreij Karpathy (Atari Pong)	2 (Effects of Null RVMs)	10	First 150
	3 (Effects of negative RVMs)	10	First 150
Dhruv Parthasarathy (Atari Pong)	4 (Effects of positive RVMs)	10	First 150

Table 3 Potential sample sizes and quantities per agent for each solution (Author 2017).

3.4.5 Data Analysis

With the project time frame being so narrow, it is important to consider the length of time it takes to generate, collect, document, analyse and present the results. As time is an important comparable factor, the results need to be compartmentalised with discreet sample sizes across a range of agents. However, there needs to be sufficient data to represent trends, compute sum, mean or running averages for comparison and account for margins of error or standard deviation. Samples have been gathered from the original solutions (Frans 2016a; Karpathy 2016a; Parthasarathy 2016) for comparing initial performance to RVMs.

3.5 METHODS

The following methods are used to accomplish objectives and serve as a potential approach to similar research projects.

3.5.1 Planning

Chronological plans with clear objectives and tasks, are initially drawn up using Microsoft Project in the form of Gantt chart to monitor project progress and facilitate the initial project proposal (see Appendix A, Section 6). Due to an overestimate of time to complete the research for a literature review, another Gantt chart was created having the time condensed to one month and the complexity of the original was also condensed with new milestones and timescales to reflect changes or anticipate new timescales and tasks (see Appendix B).

A project flow chart has also been created during the literature review serving as a functional overview of tasks to accompany the project proposal. This dissertation has also been organised logically from the beginning, with clear chapters and subheadings.

3.5.2 Literature review and research

Relevant secondary resources are obtained from the university or community libraries, academic papers and information hosted online from international experts or organisations. The resources are evaluated for suitability and are used to enhance prior knowledge, skills and learn from new or current ideas.

These resources are also used to establish previous and current research providing an insight to potential solutions applicable to this dissertation and allows for background information of the research. Arguments of topics have been refuted or concluded with introspect and aide the course and shape of the dissertation.

3.5.3 Establishing and eliciting requirements

From the resources gathered and information presented by international experts, requirements are elicited to accompany the long and short-term aims of the project and have been assessed for their suitability to this dissertation though, there is more research required long-term.

3.5.4 System specifications

Appendix D includes a system specification aiding replication of testing environments used in this project. There have been issues when trying to install the toolkit using a Microsoft Windows system, hence a Linux distribution was used instead. Most other work has been produced using industry standard file formats and Microsoft tools.

3.5.5 Design

Since limited experience for this type of research is a risk to project success, other known solutions have been used as a starting point (Frans 2016a; Karpathy 2016a; Parthasarathy 2016). As a result, the designs (Appendix D, D.2.3) elaborate on RVM exploration process and establishing how this may be achieved.

Reward values appear to be defined in the environment and normalised to real and or normalised to infinite values by the back-end of gym toolkit (OpenAI 2016b). Since the gym toolkit does not normally permit modification of reward values without extensive modification (discussed later), the

design has become more a process of how, rather than what. To simplify RVM, several different sets of values have been defined for comparison (see Table 4).

Reward Value/Type	*Neutral*	*Negative*	*Positive*
High	+/-200	-200	200
Medium	+/-100	-100	100
Low	+/-10	-10	10
Original	0	-Infinite or -1	Infinite or 1

Table 4 Shows potential and current values for each reward type (Author 2017).

While exploring, anomalies were observed during the result gathering phase. After brief discussion with the project supervisor, it was established the root cause may be random seeding. Reactive discussion was held around accommodating for this anomaly.

As there are time limitations, the data collection plan was adapted for another phase of result generation. The new phase takes account of anomalies from seeding and collect a multiple range of samples for each agent and their respective RVMs aiming to quantify running mean average, best and worst case for each agent and try to establish deviation (see Appendix D, D.3).

3.5.6 Data Integrity

In Computing, a seed is random number series in a fixed range which will produce the same series and a random starting point each time the program or function is run (Henderson 2009, p.399). The purpose of the seed ensures the same random series across implementations (OpenAI 2016b).

In RL, seeds introduce and may help reproduce different environment dynamics and provide robust implementation development and evaluation. However, seeds can be specified by individual developers as such, include the seed used (if at all) when submitting results for official evaluation or replication.

Factoring in randomness from seeding, several runs of the same implementation with different seed values need to be made to gather a representative sample of result. Once the results have been gathered for one implementation, the same process (Appendix D.3.1) should continue for the next implementation.

This project yields ten sets of 150 episodes for original implementations and any subsequent RVMs (see Appendix D.3.3 and D.3.4). Collectively, results provide sum of cumulative reward and time taken for episodes, mean average, worst and best case for every implementation.

3.5.7 Implementation

Existing solutions from other developers are used to test the RVMs and their effect on an agent' actions in an environment. In addition, other modifications are made to account for timing since time is an important and desirable comparison. In example, the arcade game Pong does not issue rewards for hitting the ball with the paddle. However, if there was a reward for this action, expected observations may be slightly longer game times as the agent learns to acquire rewards for hitting the ball.

Since rewards appear to be dispensed by the back-end toolkit, establishing how to introduce RVMs for the implementation phase is key to this project. After other avenues were explored and exhausted, RVMs were implemented within the main method of current solutions as shown for example in Chapter 5.

3.5.8 Results

The results are tabularised in a spreadsheet and presented numerically (see Appendix D.4). For each original solution, without much modification and without any reward modifications, the results are gathered. This provides the original benchmark to compare against. The RVMs and results are presented in separate files both in the python scripts (.py) and recorded in notebooks (.txt) and spreadsheets (.xlsx). For each run, the method in Appendix D.3.1 should be followed to account for seeding.

3.5.9 Backups

All work and results are backed up to five separate locations. Three different systems (One production) synchronised via remote online file storage and an external hard disk, totalling two remote locations. All work including this dissertation will be backed up to CD/DVD ROM and accompany the printed versions of this dissertation. All other backups will be archived or removed.

4 DESIGN

4.1 OVERVIEW

The designs and methods associated with them (Chapter 3), reflect the planned process utilised in completing this project.

4.2 EXPLANATION AND JUSTIFICATION

An initial aim to establish the effects of RVMs on agents has considerably affected the design process. No large-scale system designs have been used since only small code blocks are required for experimentation. An initial establishment of attainable requirements are drafted to supplement the RVM designs in Appendix D.3.3 and consideration of the data required for analysis and representation were produced following planning stage. Appendix D.2.1 is a low-level abstraction of the projects intended flow and Appendix D.2.2 highlights an adaptation for immediate rewards from Abbeel and Schulman (2016) and Silver (2015a).

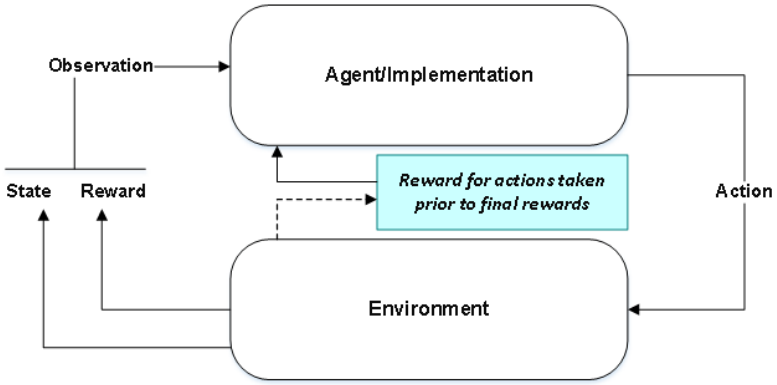


Figure 7 State transition with modified rewards (Author 2017; Adapted from Abbeel & Schulman 2016; Silver 2015a).

Visual designs (Appendix D.2 and D.3) were not created until after an RVM implementation was successfully achieved since time was of the essence. After an RVM within the main method was established successful, a hybrid qualitative-quantitative data gathering processes was designed to maintain and represent control (see Appendix D.3.1) shortly followed by a flow for the entire collection and representation phase in Appendix D.3.2.

After observing seeding anomalies, a third project plan had to be considered to factor in an additional timeframe for secondary data collection and representation which had to be scheduled whilst continuing the write-up to ensure the project remained on schedule (see Appendix B.1.2).

Discreet sets of data have been generated and collected to accommodate time limitations however, sufficient to infer initial trends of performance and with enough samples (first 150 episodes in each set) to account for randomness and deviation (see Appendix D.3.4).

With the design and implementation stages restricted by project milestones, it had not been possible to visually represent the designs or modify the project timeline prior to implementation or issues encountered. Reflective stages were held during meetings with project supervisor taking a reactive approach, facilitating circumstantial adaptation. In addition to maintaining the project Gantt with progressive updates to ensure project control, informal secondary reflection had been documented in a project diary (see the attached CD/DVD) to maintain an introspective record of successes, difficulties, failures and project progress.

5 IMPLEMENTATION

5.1 OVERVIEW

The implementation consists of exploring and establishing avenues for RVMs to known solutions and examine behavioural differences in modified RL agents. This chapter explains technical aspects of the implementation.

5.2 CURRENT KNOWN SOLUTION OVERVIEW

Frans (2016b) Hill-Climbing solution for CartPole (Figure 8) was used as a testbed for exploring the effect of RVM implementations and monitoring an agent' response to the potential modification method (Appendix D.2.3).

```
#HIll-Climbing algorithm: Every episode, add some noise to the weights, and
# keep the new weights if the agent improves

noise_scaling = 0.1
parameters = np.random.rand(4) * 2 - 1
bestreward = 0
for _ in xrange(10000):
    newparams = parameters + (np.random.rand(4) * 2 - 1)*noise_scaling
    reward = 0
for _ in xrange(episodes_per_update):
    run = run_episode(env,newparams)
    reward += run
    if reward > bestreward:
        bestreward = reward
        parameters = newparams
        if reward == 200:
            break
```

Figure 8 CEM algorithm implementation (Frans 2016b).

Frans solution (2016a) starts with values low or random values and increments the value with noise-scaling until a desired optimal solution is reached. Parthasarathy (2016) and Karpathy' (2016a) Atari Pong solutions are effectively similar, two reference images are desaturated, the backgrounds removed and the images are down sampled.

The two dimensional NN matrix is fed images, computes the difference between the two reference images. Then processes them to compute probability of moving one direction or another and with policy-gradient (PG), uses weighted probability to learn which actions will be optimal in beating the opponent (see Figure 9).

Karpathy (2016b) highlights the strength of cross-entropy method (CEM) as opposed to PG which can take a long time to train. CEM is the hill-climbing method used by Frans (2016b), initial observations after successful RVMs, reflect performance differences between PG and CEM.

```
# stack together all inputs, hidden states, action gradients, and rewards for this episode
epx = np.vstack(xs)
eph = np.vstack(hs)
epdlogp = np.vstack(dlogps)
epr = np.vstack(drs)
xs,hs,dlogps,drs = [],[],[],[] # reset array memory

# compute the discounted reward backwards through time
discounted_epr = discount_rewards(epr)
# standardize the rewards to be unit normal (helps control the gradient estimator
variance)
discounted_epr -= np.mean(discounted_epr)
discounted_epr /= np.std(discounted_epr)

epdlogp *= discounted_epr # modulate the gradient with advantage (PG magic happens right
here.)
grad = policy_backward(eph, epdlogp)
for k in model: grad_buffer[k] += grad[k] # accumulate grad over batch

# perform rmsprop parameter update every batch_size episodes
if episode_number % batch_size == 0:
  for k,v in model.iteritems():
    g = grad_buffer[k] # gradient
    rmsprop_cache[k] = decay_rate * rmsprop_cache[k] + (1 - decay_rate) * g**2
    model[k] += learning_rate * g / (np.sqrt(rmsprop_cache[k]) + 1e-4)
    grad_buffer[k] = np.zeros_like(v) # reset batch gradient buffer
```

Figure 9 PG algorithm implementation (Karpathy 2016a).

5.3 RVM EXPLORATION

An exploratory approach to research and designing RVM implementations was taken, where several crude attempts revealed one immediately plausible possibility.

5.3.1 Backend

Examining the environment exposed the rewards to be issued from the back-end either from the individual environments themselves or from the Gym toolkit (see Figure 10). The focus thereafter, was to allocate new values to the back-end reward functions or replace them.

```
import gym
import numpy as np
from gym import envs
from gym import RewardWrapper
env = gym.make('CartPole-v0')
render = True

print env.reward_range
```

```
[2017-03-09 23:12:52,690] Making new env: CartPole-v0
(-inf, inf)
```

Figure 10 Examining the rewards via print function, notice two infinite values.

An initial attempt to modify these values was implemented (see Figure 11) however, this did not yield any observable changes in reward values output in the console.

```
""" #new_reward : If permitted to access the reward_range object and modify
it from gym.core, theoretically this would be possible. """

def new_reward():
    reward = env.reward_range(-10, 20) #RESULT: cannot call the object from gym.core
    return new_reward(reward)

""" TEST FAILED """
```

Figure 11 Failed attempt at allocating values to rewards (Author 2017).

5.3.2 Function

The purpose of the function was to create an RVM which could be plugged into existing code, Figure 12 depicts a second attempt. On review, it is immediately apparent why this block will not work. Integers assigned to infinite values will have no effective change on an infinite value, on closer inspection the `reward_range()` has not been allocated the rewards from the functions arguments `def new_reward(reward)`, instead values have been allocated then assigned to `reward`.

The reward cannot be called in the for loop from `reward_range()` since it does not exist as an assignment. In hindsight, the first attempt is closer to effecting behaviour than the second. Given more time and attention, these two variants and subsequent RVM attempts can be improved.

```
#new_reward function
def new_reward(reward):

    """ Adjust the value positively or negatively (Theory of severe punishment) or enhance
    rewards. """

    reward_range = (-np.inf(-10), np.inf(2))
    reward = reward_range()

    for reward in reward_range:
        if reward < 0:
            reward = -np.inf
        elif reward == 0: #Since rewards can be gained or removed for some action (Pong is 0
        for hitting the ball).
            reward = np.inf #We can modify the reward value for hitting the ball to improve
            chances of learning (hopefully).
        else:
            reward = -np.inf
        new_reward = reward
    return new_reward
```

Figure 12 A crude RVM function (Author 2017).

5.3.3 Wrapper

Exploring ‘RewardWrapper’ import from Gym core (OpenAI 2016b) presents two options, either direct from core.py (see Figure 13) or implementing a separate environment wrapper. Both options were explored however, again presented no change to reward values.

```
class RewardWrapper(Wrapper):
    def _step(self, action):
        observation, reward, done, info = self.env.step(action)
        return observation, self.reward(reward), done, info

    def reward(self, reward):
        neg_value = -10
        pos_value = 5
        zero_value = 1
        if reward == 0.0 and not 1.0:
            reward = zero_value
        elif reward < 0.0 and not -1.0:
            reward = neg_value
        else:
            reward = pos_value
        return self._reward(reward)

    def _reward(self, reward):
        raise NotImplementedError
```

Figure 13 Rudimentary RVM to core.py (Author 2017).

5.3.4 Main method

Exploring RVM implemented in the main method, presented evidence of behavioural change though, not before changing focus from the reward values output in the console to the observable behaviour in the rendered environment, discussed later.

```
for reward in xrange(200):
    if reward == None or 0:
        rewards = 0
        reward = rewards
    elif reward < 0:
        rewards = -200
        reward = rewards
    else:
        rewards = 10
        reward = rewards
```

Figure 14 RVM for loop with if statements from Frans CEM (Author 2017).

5.4 KEEPING TRACK OF TIME

An implementation for timing and comparing algorithm performance had to be decided upon, either using the clock time or CPU time. The processor clock method was decided as it was quicker to integrate a precise timing solution (see Figure 15).

After importing the time function, `game_count` is later used as a divisible; `time0` is used as timer starter, `time1` is a stopping point to get time per game where `time2` is dual purpose for stopping the timer and as episode time. A mean is calculated for each game in an episode and the `game_count` is reset.

```
import time as t
```


```
game_count = 0
```


```
time0 = t.clock()
action = 2 if np.random.uniform() < aprob else 3 # roll the dice!
```


```
if reward != 0: # Pong has either +1 or -1 reward exactly when game ends.
  game_count +=1
  time1 = t.clock()
  total_time = ((time0-time1)*10)
  #print('ep %d: game finished, Time: %f seconds. reward: %f' % (episode_number,
  total time, reward)) + ('' if reward == -1 else ' !!!!!!!!')
```


```
time2 = t.clock()
episode_time = ((time0-time2)*10)
mean_time = episode_time/game_count
game_count = 0
```

Figure 15 Timing implementation factoring in game time, episode time and average time per game in an episode (Author 2017).

5.5 RVM IMPLEMENTATION

After exploring potential avenues for RVM, several timed RVM prototypes were created for each known solution.

Initially the reward values are assigned randomly and results gathered in discreet counterparts to compare and observe patterns however, behavioural anomalies were observed due to seeding. Several other prototypes were created using controlled RVMs (Appendix D.3.3) however, due to project schedule, only high and original values have been used during second results generation.

6 RESULTS AND DISCUSSION

6.1 OVERVIEW

The results and discussion provide broad overviews of the project successes, difficulties and failures and includes some interesting observations. Results in these examples have been discussed in greater depth through the technical report in Appendix E.

6.2 FINDINGS – AN OVERVIEW

CEM results (Frans 2016a), reflect better performance for negative RVMs, the Pole is balanced on the Cart for longer periods of time and reward average and reward frequencies improve above the original implementation (Appendix D4). In addition, the RVM behaviour of CartPole adapt better to changes in the environment, compared to the original (see Figure 16). Figure 16 is a visual example of the movements in the environment, the vertical Y-Axis represents units of movement and the horizontal X-Axis is a count of episodes. A better resolution can be found on the attached CD/DVD.

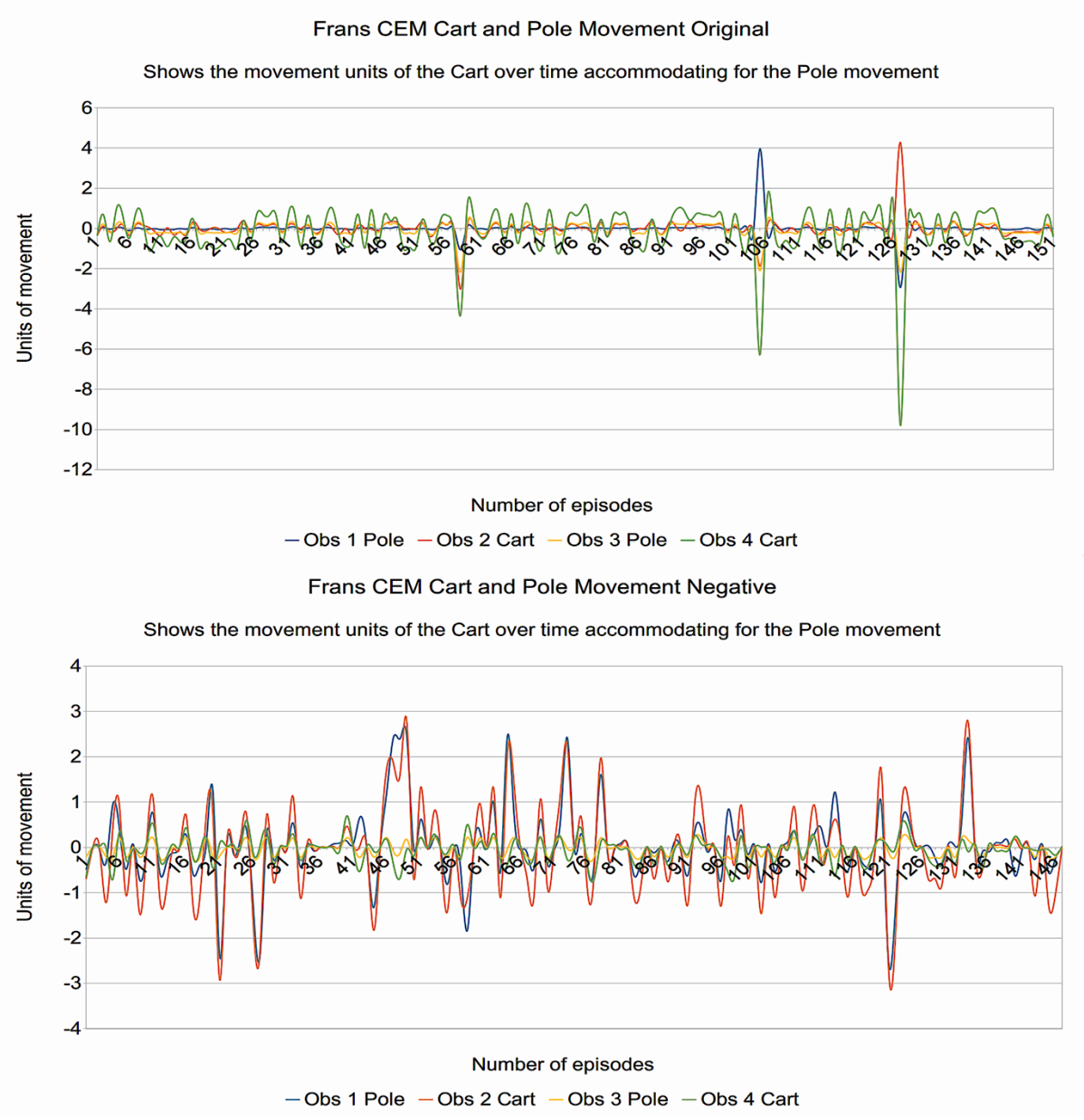


Figure 16 Comparison of environmental adaptation for original and high negative RVM

However, PGs implemented by Karpathy (2016a) and Parthasarathy (2016) do not necessarily reflect the same volumes of increased performance. Anomalies observed with Parthasarathy' solution seem to be a result of the epsilon (discussed in Appendix E), where the modified value had not been accounted for and effected the PG updates in the NN. Thus, the results observed are very different to the trends observed between Frans CEM and Karpathy PG (see Figures 17 to 19).

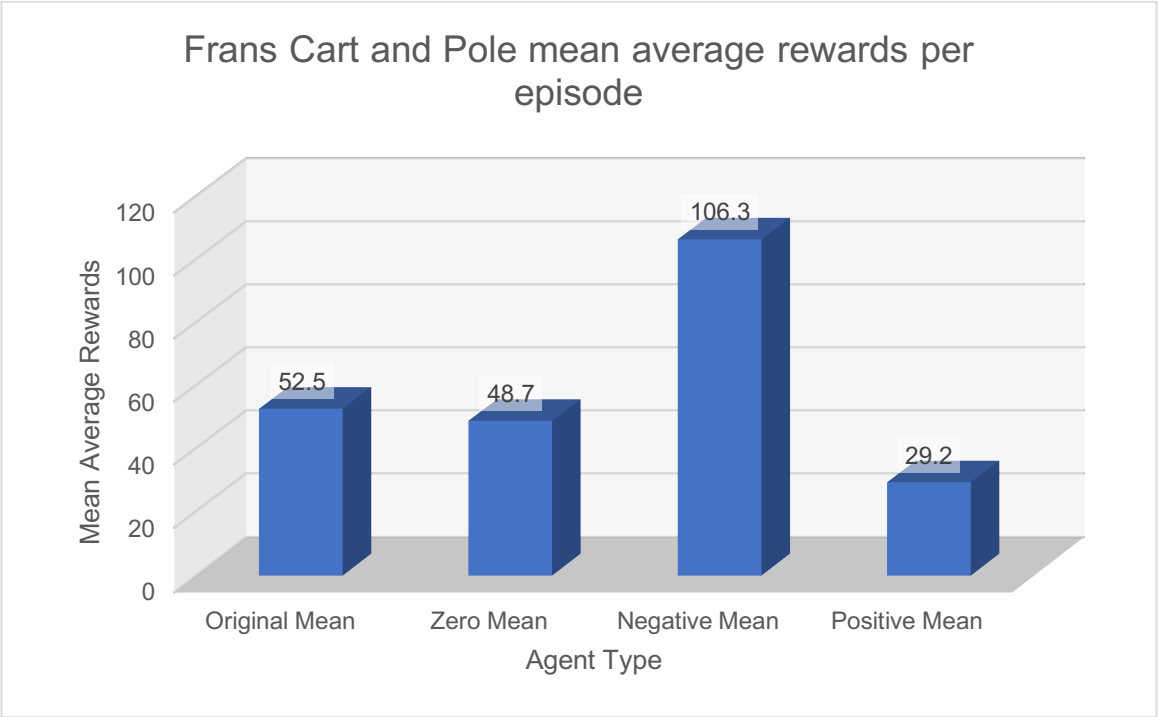


Figure 17 Shows average rewards accumulated for Frans CEM across 1,500 episodes for each RVM and the original.

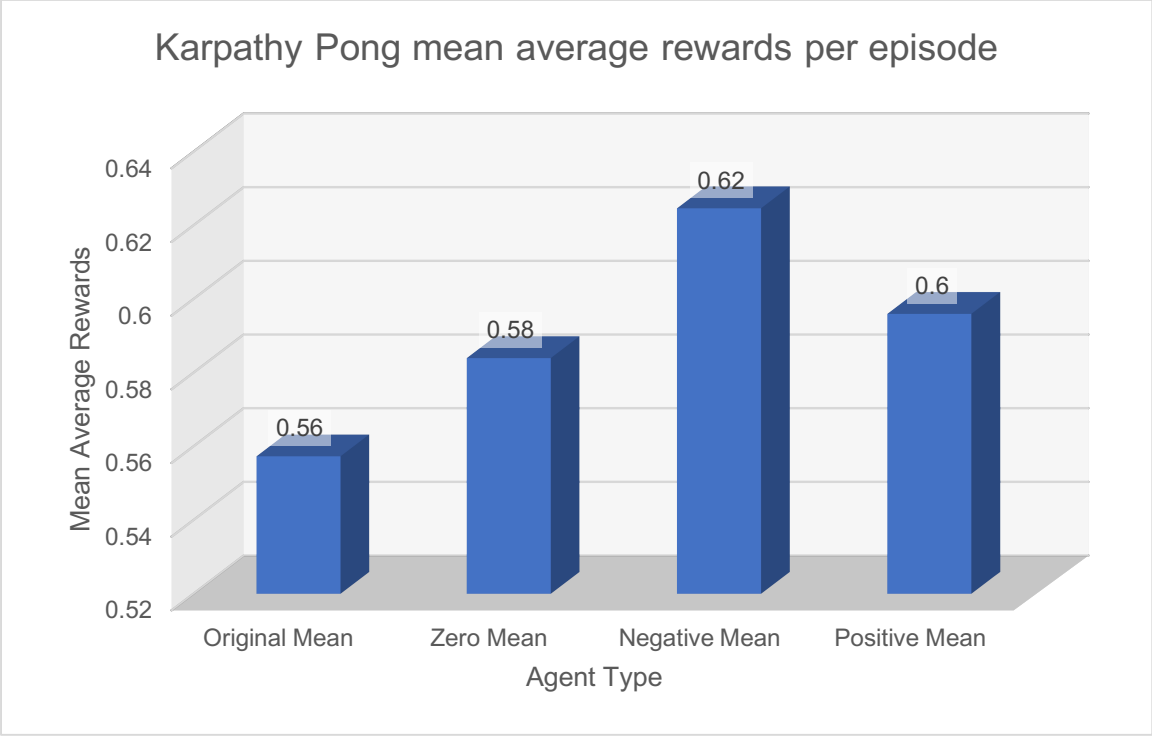


Figure 18 Shows average rewards accumulated for Karpathy PG across 1,500 episodes for each RVM and the original.

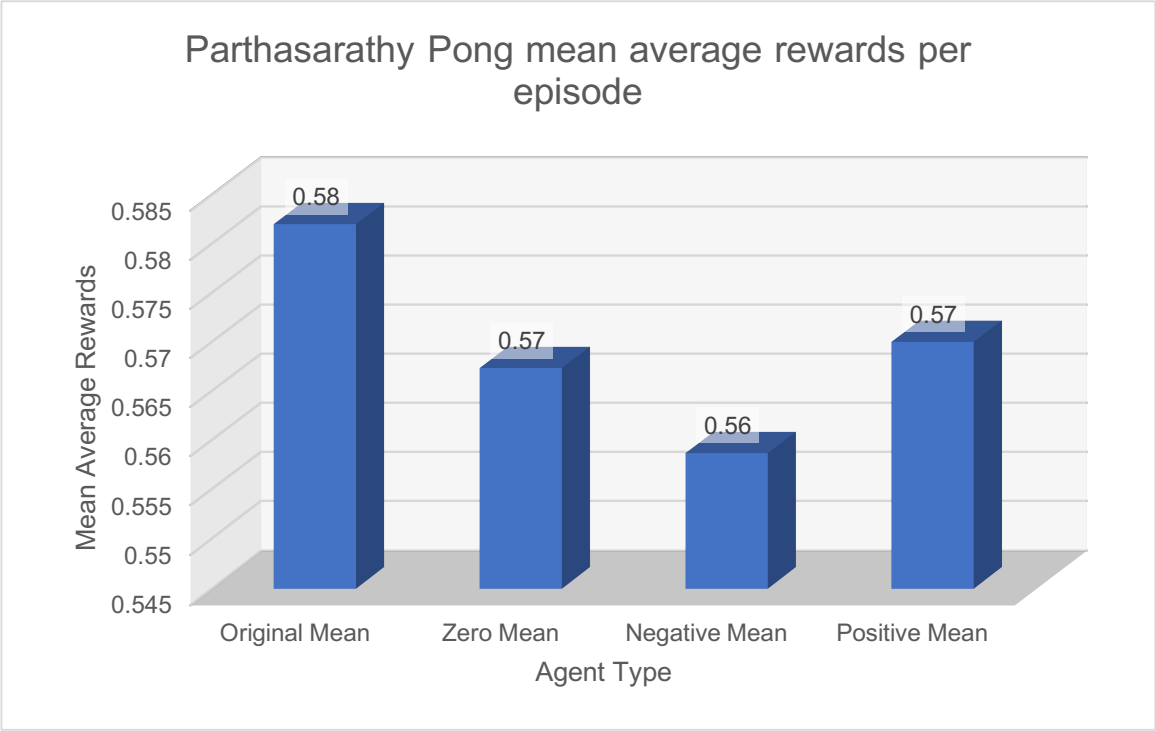


Figure 19 Shows average rewards accumulated for Parthasarathy PG across 1,500 episodes for each RVM and the original.

Due to inconsistencies between PG algorithms, whether RVMs positively effect agent behaviour is arguable, particularly in the infancy of this research. Furthermore, adding RVM to zero values has not produced slightly longer game times as predicted for the Pong environment (see Figures 20 and 21). To test this prediction, discreet samples of episodes from both agents when the paddle hits the ball will need to be collected for comparison.

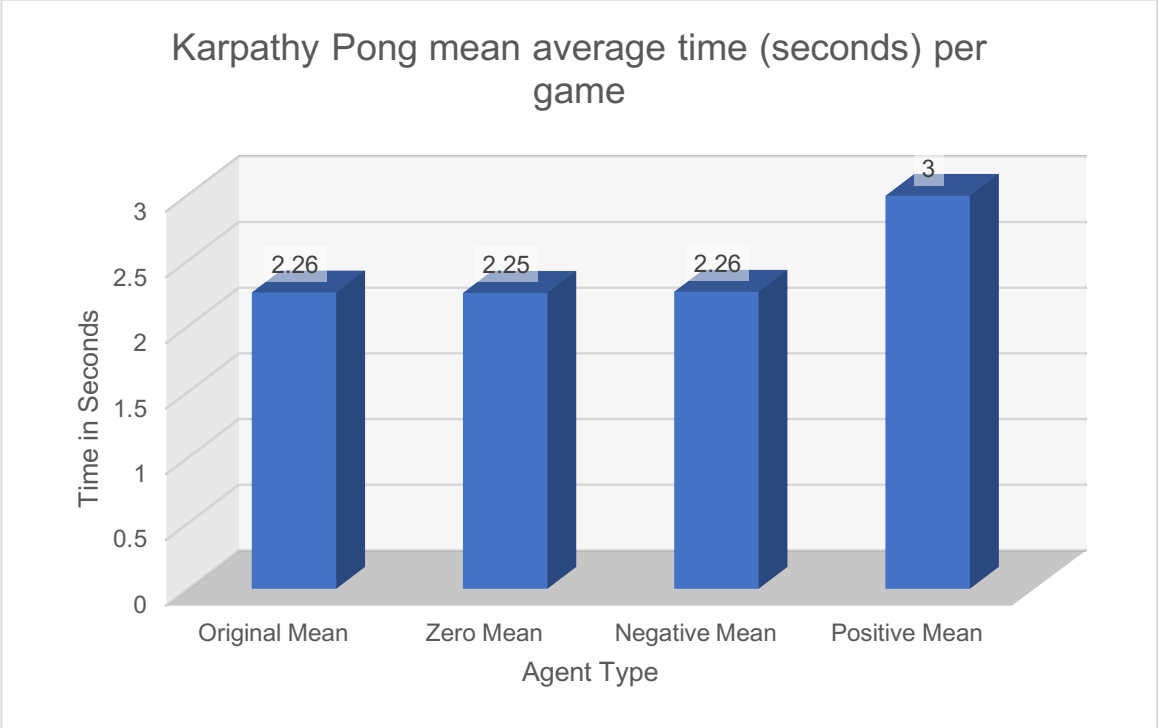


Figure 20 Average game time across 1,500 episodes of Pong using Karpathy' PG, zero or 'neutral' has not exceeded Original game time.

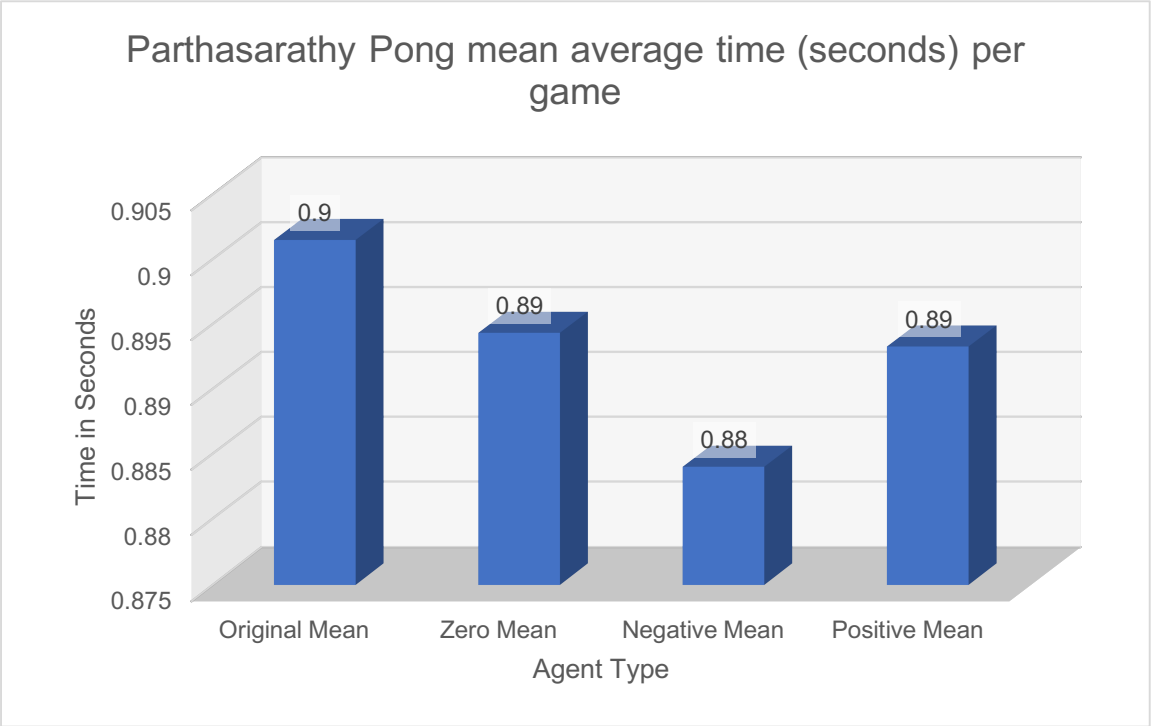


Figure 21 Average game time across 1,500 episodes of Pong using Parthasarathy' PG zero or 'neutral' has not exceeded Original game time.

6.3 Discussion

6.3.1 Trends and Performance

Observing trend anomalies from Parthasarathy RVMs has prompted another ephemeral gathering phase. After resetting epsilon value (1e-5), 3 simulations for original and negative RVMs have been collected establishing whether trends observed between Karpathy and Frans RVMs, are true to Parthasarathy since resetting the epsilon (see Figure 22).

Negative RVMs statistically outperform the original counterparts when reviewing the frequency of reward accumulation (Appendix D.4.1) with this trend also exhibited across most RVMs for PG. Sufficient to say, for the CEM and PG implementations tested, large negative rewards positively affect an agent' performance and given more time to generate another 10 sets per RVM for Parthasarathy, we may expect to see similar trends.

In addition, across most implementations for negative RVMs, some standard deviations are lower exhibiting increased consistency compared to deviation for original implementations discussed in Appendix E. Furthermore, results from this project exhibit promising statistics for increased efficiency from negative RVMs however, similar trends may not be apparent in other algorithms or across longer testing periods.

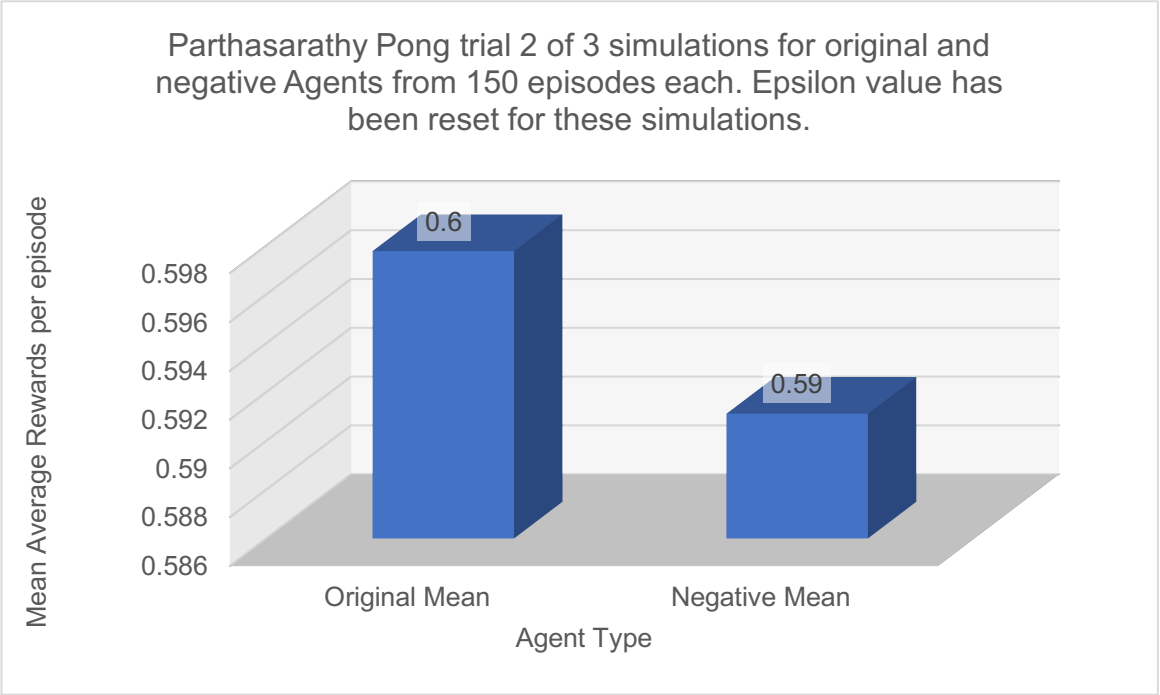


Figure 22 Compares original and negative RVM across 3 simulations of 150 episodes per agent with marked improvements.

6.3.2 Scalability

Without sufficient volumes of evidence supporting the hypotheses, scalable solutions cannot be reduced to an algorithm since the volume of data collected in this project only spans across a short time period with minimal variance in both task and agent.

Without the volumes of data required, patterns and inconsistencies cannot validate or verify the results collected nor account for anomalies which may affect safety and ethical behaviour when introduced into society. Thus, producing a viable algorithm with immediate and long-term reward combinations would be premature.

Furthermore, to evaluate potential scalability, Agents must first be tested in higher complexity environments, sandboxed online environments and robotics; none of which can proceed without first definitively verifying, RVMs improve agent performance and capabilities safely, and within ethical and legislative boundaries.

6.3.3 Ethics and Safety

Although fictional, Isaac Asimov introduced four laws of robotics with the Zeroth being appended to supersede the first three:

Zeroth. A robot may not harm humanity, or, by inaction, allow humanity to come to harm.

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.

2. A robot must obey the orders given to it by human beings, except where such orders would conflict with the First Law.

3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.

Figure 23 Isaac Asimov's three laws of Robotics (Amended from Barthelmess & Furbach, 2014).

Solutions are engineered not with the laws of Robotics but with similar precautionary measures built-in to satisfy safety and ethical considerations. Autonomous automobiles are one example where rules synonymous to the ‘three laws’ are applicable for the safety of the consumer and others on the road.

Behaviours exhibited during this project are still erratic at best and although the scale of standard deviation is low across all discreet samples. In larger samples or engineered deployments a 0.0000000001 percentile deviation, could be the difference in autonomous machines performing as expected or unpredictably compromising safety.

Additionally, observing rendered Pong (PG) outputs do not yet exhibit improved interaction compared to Frans CEM though, statistics show otherwise. In isolation, RVM improvements may not satisfy safety and ethical considerations while exhibiting unpredictable behaviours however, once trained this inference may prove incorrect.

Ideally, an algorithm which does incorporate immediate and long-term rewards whilst satisfying all core AI safety requirements (Brockman & Christiano 2016) and performs predictably at a whole range of tasks, would be deemed safe. However, the ethics involved in such a solution may not be welcomed by society if job loss or economic issues were to occur for example.

Moreover, though increased performance is exhibited from agents with negative RVMs, this project has not explored specific behaviours used to gain performance. Reiterating a previous example, if the performance increase was attained from committing robbery to upgrade hardware, this is neither an ethical or lawful action and introduces risks.

7 CONCLUSIONS

7.1 SUMMARY

Since there are no rules for how autonomous machines should behave (Wakefield 2015), the importance of investigating behaviours in ML Agents and their responses to inhibitors and activators; is as critical to progressing the development of AI systems, as it is to be standardising the engineering, legislative and safety precautions required prior to mechanical autonomy. Without such research, elements of risk are introduced into society, economy and industrial work places when autonomous systems are used.

RL Agents possess no prior knowledge (unless trained in advance) of the environment it will be acting in and takes an empirical approach to establish and learn environment boundaries and its own limitations. RL utilises a combination of delayed reward signal and probability of actions to maximise expected cumulative reward as a goal (Silver 2015a, 2015b). Considering AI is "*an ambiguous model (similar to human reasoning)*" (Schalkoff 1990, p.xxii), delaying the reward signal or sacrificing rewards is an unexpected approach to reward signalling in ML agents, particularly since animals may prefer immediate rewards over long-term though, the current approach to RL improves long-term performance.

Exploring a combination of delayed rewards and immediate rewards by introducing RVMs into Agents, harvests some astonishing results. Comparing original implementations from Karpathy PG (2016b) and Frans CEM (2016a) to high value reward inhibitors, has improved reward gain with differences of 67 and 101 percent respectively. In addition to increased rewards, the reward and time standard deviation across 1,500 episodes, has improved by as much as 2.5% for CEM, representing higher consistency in desirable actions. Considering a 0.0000000001 percentile deviation, could be the difference in correct or unpredictable behaviour from an autonomous machine, this is a huge improvement.

Since high value positive RVMs have not produced such promising improvements across all three implementations compared to high reward for negative actions, the consequence of an RL agent using negative actions and being ‘punished’ for them, demonstrates higher significance to improving performance. This effect would be desirable in dynamic environments, where actions must be learned quickly, though this is not necessarily safe.

7.2 EVALUATION

7.2.1 Objective 1

An empirical methodology has been used to establish and deploy RVMs into existing RL solutions. The exploration associated with empiricism, proved to be uncontrolled initially and consumed the vast majority of time allocated during the implementation phase, reducing allocated time to graphically represent designs, later produced during data generation and representation stages.

Quantitative designs for collection methods and data generation, have been used to complement qualitative methods, utilised for controlled collection of results in discreet counterparts from each prototype. Though mistakes have been made through various incremental phases of each prototype, reactive discussions and further exploration prompted corrective decisions to remedy issues encountered and new results were generated where time limitations permitted.

7.2.2 Objective 2

The anomalies exhibited by Parthasarathy' implementation (2016) were caused by an overlooked modification to an epsilon value (controls PG update value) during exploration stages. Unfortunately, insufficient time remained to generate enough results to confirm the trends exhibited by other RVMs across other implementations however, a small collection shows promising trends (Appendix D.4.5) when compared to results in Appendix D.4.4.

With project success measured on an RVM implementation being successfully deployed and an RVM exhibiting higher performance, the RVMs deployed in Frans CEM (2016a) and Karpathy' PG (2016a) implementations, have met this criterion successfully.

Negative RVMs provided higher average reward and reward frequencies, more consistent behaviour measured through standard deviation is reflected and though the average time per episode was higher in each negative RVM, the Agent is successfully acting on the environment for longer periods of time thus, preventing the episode from ending by winning more games of Pong against the opponent, for example.

7.3 FUTURE WORK

Time limitations have inhibited the potential volume of current RL implementations which could have been used to test RVM deployments, narrowing the volume of data required to reflect trends otherwise exhibited in larger counterparts of results. Time restrictions also prevented additional studies on a larger range of algorithms and environments with varying tasks subsets and complexities. In gathering and quantifying more information, conclusive evidence for RVM effects are produced providing greater understanding of behavioural AI for application in other controlled experiments.

In addition, longer experimentation and testing times will yield more results enabling better statistical accuracy and will highlight unseen anomalies in preparation for deployment in areas of human economy moving forward. Out-dated research can continue to be rendered obsolete in favour of new methods as understanding improves. As a direct result, other studies of this nature ease the refinement and elicitation of requirements, which will aid the standardisation for hardware development and compliment the engineering of future mechanical solutions.

Since the scope of this project does not explore NN complexities and considering both Karpathy and Parthasarathy have used 2 layered NNs in their implementation, the effects of RVM on NN weights in other NN types should also be explored. In particular, where both algorithms and NNs are used in unity (Agent) when solving problem spaces involving complex real-world or simulated dynamic environments.

Even though Agents with negative RVM solutions have exhibited performance improvements, it is not clear how the combination of maximising expected long-term rewards and immediate rewards for actions would behave, coupled with robotics. To satisfy or refute the hypotheses in their entirety, further controlled exploration and experimentation is still required. It will be interesting to see how immediate and long-term combinations (ILRC) will behave when combined with a group of NNs designed to tackle many tasks, particularly where each NN deals with emulated human cognition (logical, lateral, creative) and observations (visual, kinetic, audible, sensory). Arguably everything humans experience, including thought, is an observation of the environment. To

simulate on a small small-scale with several NNs is one project, the second is to test the ILRC on the cognitive NN.

The types of developments and proposed research presented, should therefore proceed with ethical and safety considerations for the foreseeable future. As comprehension of autonomy and socio-economic precautions develop in unity with cohesive progression, improvements to engineering standards and legislation frameworks will support safe integration of autonomous systems into society and humanity will be better equipped to prepare for what resides on the horizon.

Word Count Including Tables: 9,996

APPENDIX A – DETAILED PROJECT PROPOSAL

Undergraduate Project Proposal Form

Degree Title: BSC (Hons) Computing.	**Student's Name: Russell Allen Eaglesfield Clarke**
	Supervisor's Name: Marcin Budka
	Project Title/Area: **Four laws in practice & Neural networks.** **Computing/Science.**

Section 1: Project Overview

1.1 Problem definition - use one sentence to summarise the problem:

As BBC Technology reporter Jane Wakefield (2015) states, "We don't currently have any rules for how robots should behave if and when they start operating autonomously"; This is the very foundation for my problem and long term research area.

1.2 Background - please provide brief background information, e.g., client:

At present this certainly is notable as an area of research. With growing concerns into the ethics of AI and robotic machines in the human form; it is evident there are concerns about autonomous machines and their capabilities if the outcome of inappropriate behaviour should manifest (EU Parliament 2016). As such the long-term area of research would lead to defining behaviour in 'intelligent' or 'autonomous' machines and the potential for hardcoding this information into hardware.

1.3 Aims and objectives – what are the aims and objectives of your project?

Investigate the behavioural nature of AI systems using values based on the fictional writer Isaac Asimov's four laws of robotics with a particular consideration of Neural Networks and 'intelligent' or 'autonomous' systems. Intention is to derive sufficient results to confirm the possibility of behavioural 'intelligence' [at a very basic level] as opposed to 'automation' in computing systems.

Assumptions:

- Clearly state and document events and intentions.
- Provide evidence and documentation where applicable.
- Seek advice where necessary.
- Meet with supervisor to discuss progress.
- Always try to adhere to ethics, standards and law (National and International where applicable). 'Try' is applicable in this circumstance due to the nature of 'change'.
- All assumptions are assumptions and do not necessarily reflect on real world situations accurately.

Research:

- Research Computer Neural-Networks, Behavioural AI and ethics surrounding them.
- Establish if there are any forms to date, of behavioural AI and to what avail or level.
- Establish and justify if they void the current project subject and provide reasoning for this.
- If there are already solutions in place, establish if they can be improved.

- Establish the current trends in Behavioural AI (if any) and the problems encountered.
- If there are not any current solutions, provide ethical reasoning and requirement for such research to continue and supply justification or evidence.
- Investigate Behavioural AI requirements and application in Neural-Net technology or any other domains and any potential standards, laws or ethics which must be adhered to (also part of **'Investigate'**).

Investigate:

- Check the potential applications of the above and their compliance with standards.
- Justify the potential applications of the above and their compliance with standards.
- Decide how to proceed and liaise with supervisor and or specialists.

Analyse:

- Analyse requirements of Behavioural AI and Neural-Nets and if there aren't any, extrapolate the requirements from the investigative and research stages.
- Clearly state and justify the requirements providing evidence.
- Check the requirements are applicable to the research.
- Check the research complies with the requirements.
- Establish an appropriate methodology and consider a safe, controlled data collection method.

Design:

- Using the appropriate methodology Hybrid SDLC, design, define and try to predict the measurements which will be used for project success rate.
- Consider how the tests should be written and if they reflect accurately on any results which could be predicted and any surprising results.
- Create a test plan for each of these prototype designs.
- Document already established evidence, methods or processes.

Implementation:

- Following the processes and in keeping with the methodology and research, begin to implement the research to extrapolate established and unestablished results. OpenAI is one such toolkit which could be used.
- Record results.
- Analyse and find patterns in the results.
- Find application if successful, if unsuccessful, refer back (above) and start again.

Test:

- Check the tests are correct and suited to the project.
- Implement the test plan and build the tests into the whole project where appropriate.
- Document the results, successes and failures.
- Justify or explain the tests.

Evaluate:

- The success of the project, its strengths, weaknesses, reason for failure or to continued success where applicable.
- Project post mortem.

Section 2: Artefact

2.1: What is the artefact that you intend to produce?

Intended artefact will be in the form of documented results from test and implementation stages and any code or formulas used to produce them finalised through a report of no more than 5,000 words (Excluding appendices) which will compare and analyse and present findings during the project.

2.2 How is your artefact actionable (i.e., routes to exploitation in the technology domain)?

AI and NNs are by definition, within the computing/computer science domain and could be applicable to any domain in which computing assists in economy, humanity, the environment or public services (dependent on application and with compliance to ethics, law and standards).

Section 3: Evaluation

1.1 How are you going to evaluate your work?

- At present and without having more information, it is difficult to define how to assess or 'evaluate' the success rate of the work. The intent is to document, evaluate and regularly seek advice and guidance from relevant supervisors or specialists where applicable.
- Validate and reference knowledge acquired via third parties.
- Ensure compliance to ethics, law and standards throughout. - Quality
- Ensure the work meets the requirements of BU and BCS Accreditations - Validation.
- Potentially submit the results for peer review - Verification.

1.2 Why is this project honours worthy?

The research area seems to be a relatively unexplored area. With consideration to the complexity and additional 'extra-curricular' learning curve required to begin this research, it would stand to reason. In addition, then having to justify, clarify and condense this research into cohesion and within a comparably small time constraint, also brings value and risks to the project and its success rate. During this project, a particular qualitative as opposed to quantitative approach, may be sought in consideration of time constraints and risks.

1.3 How does this project relate to your degree title outcomes?

Through projects such as this, it is possible to expand on previously gained skills and push to specialist areas of study. In relation to degree title outcomes, improving fundamental skills in computing is an essential element in a computing degree, aids the progression from a core foundation of knowledge acquired at degree level and paves the way for continued development of these skills. It is through project based work (amongst other work), capability and accreditation can be assessed.

1.4 How does your project meet the BCS Undergraduate Project Requirements?

Through conforming to standards and guidelines outlined within, in addition to other standards, ethics and legislation set out by other organisations intended to define a specific level of quality(s) or expectation(s).

1.5 What are the risks in this project and how are you going to manage them?

- Time constraints will be outlined in a Gantt chart and there are always risks of project failure due to time constraints (among others) which shall be managed and assessed with regular visits and communication with supervisors.
- Costs are not a risk to the project at this stage. Any costs or risks involved, where the responsibilities are not covered by third party policies, terms, conditions or obligations, will be during potential visits to external laboratories or venues throughout the project.
- The risks of non-conformity to ethics, standards and law as yet, are not currently known on the basis this is a relatively new area of research.
- There are other risks such as mitigating circumstances, electrical faults and environmental factors some of which are out of human control or in the control of third parties.
- These risks are to be managed using 'common sense' where appropriate, guidelines set out in the standards and ethics research stage, time management, appropriate and professional behaviour and conformity to national and/or international law.
- Additionally, reporting any circumstance which may impede on the above or put others at risk or to harm and completing additional ethics checklists where appropriate for approval of continued research.

Section 4: References

4.1 Please provide references if you have used any.

Wakefield. J., 2015. *Intelligent Machines: Do we really need to fear AI?* [online]. Available from: [Accessed 24 September 2016].

European Parliament, Committee on Legal Affairs., 2016. DRAFT REPORT: With recommendations to the Commission on Civil Law Rules on Robotics (2015/2103(INL)) [online]. Available from: [Accessed 25 September 2016].

BCS., 2016. *Academic Accreditation* [online]. Available from: [Accessed 26 September 2016].

Section 5: Ethics

5.1 Have you submitted the ethics checklist to your supervisor? **Yes**

5.2 Has the checklist been approved by your supervisor? **Yes**

Section 6: Proposed Plan

Task Name	Duration	Start	Finish	Predecessors
Research period	10 days	Tue 18/10/16	Sun 30/10/16	
AI: Ethics, standards and legislation	3 days	Tue 18/10/16	Thu 20/10/16	
Neurological Computer Networks: Ethics, standards and legislation	3 days	Thu 20/10/16	Sat 22/10/16	
Behavioural AI: Ethics, standards and legislation	3 days	Sun 23/10/16	Tue 25/10/16	
The application to real world problems	3 days	Wed 26/10/16	Fri 28/10/16	
Write up Introduction / Hypothesis ready for dissertation	2 days	Sat 29/10/16	Sun 30/10/16	
Liaise with Supervisor - Address and discuss any Ethics, Standardisation or Legislation surrounding subject matter. Address any concerns.	2 days	Mon 31/10/16	Tue 01/11/16	1,2,3,4,5,6
Establish if there are any forms to date, of behavioural AI and to what avail or level.	5 days	Tue 01/11/16	Sun 06/11/16	7
Forms of AI, Behavioural AI and Neuro-Nets	2 days	Tue 01/11/16	Wed 02/11/16	
How they are applicable and in which domains	2 days	Thu 03/11/16	Fri 04/11/16	
How they are measured, tested and assessed	2 days	Fri 04/11/16	Sun 06/11/16	
Establish and justify if they void the current project subject and provide reasoning for this - Appointment with Marcin Tuesday?	9 days	Mon 07/11/16	Thu 17/11/16	8,9,10,11
If there are already solutions in place, establish if they can be improved	3 days	Mon 07/11/16	Wed 09/11/16	
The current trends in these areas of research (if any) and the problems encountered	3 days	Thu 10/11/16	Sat 12/11/16	
If no current solutions, provide ethical reasoning and requirement for such research to continue and supply justification or evidence	3 days	Sun 13/11/16	Tue 15/11/16	
Consider taking the project into a similar but conceptually different direction	4 days	Mon 14/11/16	Thu 17/11/16	
Liaise with Supervisor - Guidence and discussion around subject matter. Address any concerns or changes.	2 days	Thu 17/11/16	Fri 18/11/16	8,9,10,11,12,1
Investigation and continued research period	12 days	Sat 19/11/16	Sun 04/12/16	17
Check the potential applications of the above and their compliance with standards.	3 days	Sat 19/11/16	Tue 22/11/16	
Justify the potential applications of the above and their compliance with standards	5 days	Wed 23/11/16	Tue 29/11/16	
Justify the potential applications of the above and their compliance with standards	2 days	Wed 30/11/16	Thu 01/12/16	
Decide how to proceed and liaise with supervisor and or specialists	2 days	Fri 02/12/16	Sun 04/12/16	
Email Supervisor - Progress report & Guidence	1 day	Mon 05/12/16	Mon 05/12/16	17,18,19,20,2
Requirements analysis period	20 days	Tue 06/12/16	Sat 31/12/16	23
Extrapolate the requirements from te investigative and research stages	3 days	Wed 07/12/16	Fri 09/12/16	
Clearly state and justify the requirements providing evidence	4 days	Sat 10/12/16	Wed 14/12/16	
Liaise with Supervisor - Discuss suitable methodology, Guidence for suitable field trip Venues and Forms required (if any).	2 days	Mon 12/12/16	Tue 13/12/16	
Check the requirements are applicable to the research	3 days	Thu 15/12/16	Sun 18/12/16	
Check the research complies with the requirements	2 days	Mon 19/12/16	Tue 20/12/16	
Establish an appropriate methodology and consider a safe, controlled data collection method	3 days	Wed 21/12/16	Fri 23/12/16	
[Extra Curricular] Potentially plan a visit to Tech Museum/Laboratory over Christmas	2 days	Sat 24/12/16	Sun 25/12/16	27
Continuation of additional research and Investigations with additional checks to requirements.	3 days	Mon 26/12/16	Wed 28/12/16	
Continue with Requirements Analysis	3 days	Thu 29/12/16	Sat 31/12/16	27
Liaise with Supervisor - Discuss further - Methodology and Holiday break / Discoveries and achievements	1 day	Tue 03/01/17	Tue 03/01/17	29,30,31,32,3
Design, Prototyping and Testing period - Experimental	23 days	Sat 28/01/17	Tue 28/02/17	1,7,8,12,17,18
Using the appropriate methodology Hybrid SDLC, design, define and try to predict the measurements which will be used for project success rate	8 days	Sat 28/01/17	Tue 07/02/17	
Consider how the tests should be written and if they reflect accurately on any results which could be predicted and any surprising results	4 days	Wed 08/02/17	Sun 12/02/17	
Liaise with Supervisor - Discuss experimentation, Failures and Successes.	2 days	Mon 13/02/17	Tue 14/02/17	
Create a test plan for each of these prototype designs	4 days	Wed 15/02/17	Mon 20/02/17	
Document already established evidence, methods or processes	5 days	Tue 21/02/17	Sun 26/02/17	
Liaise with Supervisor - Results?	2 days	Mon 27/02/17	Tue 28/02/17	
Implementation and Testing period - Experimental	24 days	Wed 01/03/17	Sun 02/04/17	35,36,37,39,4
Following the processes and in keeping with the methodology and research, begin to implement the research to extrapolate established and unestablished results	4 days	Wed 01/03/17	Sat 04/03/17	
Check the tests are correct and suited to the project	2 days	Sat 04/03/17	Sun 05/03/17	
Liaise with Supervisor - Results, Struggles & Guidence	2 days	Mon 06/03/17	Tue 07/03/17	
Implement the test plan and build the tests into the whole project where appropriate	5 days	Wed 08/03/17	Tue 14/03/17	
Record results	10 days	Wed 01/03/17	Tue 14/03/17	
Analyse and find patterns in the results	4 days	Wed 15/03/17	Sun 19/03/17	
Liaise with Supervisor	2 days	Mon 20/03/17	Tue 21/03/17	
Document the results, successes and failures	3 days	Tue 21/03/17	Thu 23/03/17	
Find application if successful, if unsuccessful, refer back (above) and start again	2 days	Fri 24/03/17	Mon 27/03/17	
Justify or explain the tests	5 days	Tue 28/03/17	Sun 02/04/17	
Liaise with Supervisor - Begin to reassess work and prepare for submission.	2 days	Mon 03/04/17	Tue 04/04/17	
Evaluation period - Critical Analysis and Report	9 days	Mon 03/04/17	Thu 13/04/17	
The success of the project, its strengths, weaknesses, reason for failure or to continued success where applicable	4 days	Mon 03/04/17	Thu 06/04/17	
Liaise with Supervisor - Request a read through	2 days	Fri 07/04/17	Mon 10/04/17	
Project post mortem	4 days	Mon 10/04/17	Thu 13/04/17	

APPENDIX B – PROJECT PLAN REVISIONS

B.1.1 REVISED PLAN AFTER OVERESTIMATING TIME REQUIRED

Task Name	Duration	Start	Finish	Predecessors
Research period	21 days	Mon 30/01/17	Mon 27/02/17	
Neural-Networks: Type, Function and application	6 days	Mon 30/01/17	Mon 06/02/17	1
Neurological Computer Networks: Ethics, standards and legislation	2 days	Tue 07/02/17	Wed 08/02/17	1,2
AI: Ethics, standards, Implementation and types. Establish how they are applicable and in which domains, how they are measured and tested	7 days	Fri 10/02/17	Sun 19/02/17	1,2,3
The application to real world problems and toolkits: Machine Learning	5 days	Tue 21/02/17	Sat 25/02/17	1,2,3,4
Summary and proof readiing - make any ammendments to Gantt justify why they were made after the research period and explain how this will be mitigated in the future. Append to Dissertation and submit to supervisor on 27th in anticipation of intermediary	2 days	Sat 25/02/17	Mon 27/02/17	1,2,3,4,5
Interim review - Hand in to Supervisor	0 days	Tue 28/02/17	Tue 28/02/17	1,2,3,4,5,6
Design period - This must be ephimeral due to time constraints and merge with analysis and requirements stage below - normally this stage would be the most depth twinned with research stage	3 days	Tue 28/02/17	Thu 02/03/17	7,1
Research methodologies and analysis techniques, establish reqirements. If there are already solutions in place, establish if they can be improved and do the write-up.	17 days	Fri 03/03/17	Mon 27/03/17	8
Begin to implement your own artifact and research into practical methods after establishing analysis and requirements and merging design with requirements.	6 days	Fri 03/03/17	Fri 10/03/17	8,9
Start to cross reference other implementations and attempt to implement them: record results as you go along - just get them working as they take a while to run and generate results.	10 days	Sat 11/03/17	Thu 23/03/17	8,9,10
Finish up any implementation and get it running - possibly with different environments - collect results and do write up of implementation ensuring all code is referenced. Explain what the code does and how it relates to other code.	2 days	Fri 24/03/17	Sat 25/03/17	8,9,10,11
Collect the results and consider any modifications which may need to be made. Make them and rerun the code - gather more results	2 days	Sat 25/03/17	Mon 27/03/17	8,9,10,11,12
Discuss results and how to represent them - Supervisor	0 days	Tue 28/03/17	Tue 28/03/17	8,10,9,11,12,1
Results analysis, formatting and write-up	14 days	Tue 28/03/17	Fri 14/04/17	7,14
Format results and analyse them - make inferences and suggestions possibly if there is enough time, implement them and repeat.	8 days	Tue 28/03/17	Thu 06/04/17	14
Clearly justify and explain differences in implementations or any changes made.	2 days	Fri 07/04/17	Mon 10/04/17	14,16
Conclusions and tidy up of anything in the project making sure all parts are correct and formatted correctly - Make up the DVD's as well.	6 days	Tue 11/04/17	Tue 18/04/17	14,16,17
Begin to writeup conclusions, successed, failures, as a discussion and merge into concrete or infered conclusions.	2 days	Tue 11/04/17	Wed 12/04/17	14,16,17,18
Finalise conclusions ensure they are just.	4 days	Thu 13/04/17	Tue 18/04/17	14,16,17,18,19
Dissertation should be almost complete and heading for formatting and Report (as part of artifact)	1 day	Wed 19/04/17	Wed 19/04/17	14,16,17,18,19
Proof read through the dissertation and write up conclusion and abstract, proof read again and ensure any formatting is complete.	2 days	Fri 21/04/17	Mon 24/04/17	7,14,21
Finish up CD's and ensure they are in a suitable format also that the contents have been included in an appendix	2 days	Fri 21/04/17	Mon 24/04/17	7,14,21
Project post mortem	5 days	Tue 25/04/17	Sun 30/04/17	
Evaluation period - Critical Analysis and Report	5 days	Tue 25/04/17	Sun 30/04/17	
The success of the project, its strengths, weaknesses, reason for failure or to continued success where applicable	5 days	Tue 25/04/17	Sun 30/04/17	
Submit to supervisor for proof read	3 days	Mon 01/05/17	Wed 03/05/17	24,25,26
Submit to printer and binders and prepare digital copies	8 days	Wed 03/05/17	Fri 12/05/17	

B.1.2 REVISED PLAN AFTER ESTABLISHING ANOMALIES

Task Name	Duration	Start	Finish	Predecessors
Research period	21 days	Mon 30/01/17	Mon 27/02/17	
Neural-Networks: Type, Function and application	6 days	Mon 30/01/17	Mon 06/02/17	1
Neurological Computer Networks: Ethics, standards and legislation	2 days	Tue 07/02/17	Wed 08/02/17	1,2
AI: Ethics, standards, Implementation and types. Establish how they are applicable and in which domains, how they are measured and tested	7 days	Fri 10/02/17	Sun 19/02/17	1,2,3
The application to real world problems and toolkits: Machine Learning	5 days	Tue 21/02/17	Sat 25/02/17	1,2,3,4
Summary and proof readiing - make any ammendments to Gantt justify why they were made after the research period and explain how this will be mitigated in the future. Append to Dissertation and submit to supervisor on 27th in anticipation of intermediary	2 days	Sat 25/02/17	Mon 27/02/17	1,2,3,4,5
Interim review - Hand in to Supervisor	0 days	Tue 28/02/17	Tue 28/02/17	1,2,3,4,5,6
Design period - This must be ephimeral due to time constraints and merge with analysis and requirements stage below - normally this stage would be the most depth twinned with research stage	3 days	Tue 28/02/17	Thu 02/03/17	7,1
Research methodologies and analysis techniques, establish reqirements. If there are already solutions in place, establish if they can be improved and do the write-up.	17 days	Fri 03/03/17	Mon 27/03/17	8
Begin to implement your own artifact and research into practical methods after establishing analysis and requirements and merging design with requirements.	6 days	Fri 03/03/17	Fri 10/03/17	8,9
Start to cross reference other implementations and attempt to implement them: record results as you go along - just get them working as they take a while to run and generate results.	10 days	Sat 11/03/17	Thu 23/03/17	8,9,10
Finish up any implementation and get it running - possibly with different environments - collect results and do write up of implementation ensuring all code is referenced. Explain what the code does and how it relates to other code.	2 days	Fri 24/03/17	Sat 25/03/17	8,9,10,11
Collect the results and consider any modifications which may need to be made. Make them and rerun the code - gather more results	2 days	Sat 25/03/17	Mon 27/03/17	8,9,10,11,12
Discuss results and how to represent them - Supervisor	1 day	Thu 30/03/17	Thu 30/03/17	8,10,9,11,12,1
Additional result collection, analysis, formatting and write-up	13 days	Thu 30/03/17	Sat 15/04/17	7,14
Format results and analyse them - make inferences and suggestions possibly if there is enough time, implement them and repeat.	13 days	Thu 30/03/17	Sat 15/04/17	14
Clearly justify and explain differences in implementations or any changes made.	2 days	Sat 15/04/17	Mon 17/04/17	14,16
Begin to writeup conclusions, successes, failures, as a discussion and merge into concrete or infered conclusions. In addition, start the artefact write-up	9 days	Mon 17/04/17	Thu 27/04/17	14,16,17,20
Finalise conclusions ensure they are just.	4 days	Tue 25/04/17	Fri 28/04/17	14,16,17,18,2(
Conclusions and tidy up of anything in the project making sure all parts are correct and formatted correctly - Make up the DVD's as well.	1 day	Fri 28/04/17	Fri 28/04/17	14,16,17
Project post mortem	4 days	Tue 25/04/17	Fri 28/04/17	
Evaluation period - Critical Analysis and Report	4 days	Tue 25/04/17	Fri 28/04/17	
The success of the project, its strengths, weaknesses, reason for failure or to continued success where applicable	4 days	Tue 25/04/17	Fri 28/04/17	
All parts of the project and dissertation should be complete, if not speak to supervisor and request extension if required. Else should be ready to hand over to supervisor for a final proof read.	2 days	Fri 28/04/17	Sun 30/04/17	14,16,17,18,1
Proof read through the dissertation and write up conclusion and abstract, proof read again and ensure any formatting is complete and any condensing or modifications suggeted by supervisor is completed..	1 day	Mon 01/05/17	Mon 01/05/17	7,14,24,21
Finish up CD's and ensure they are in a suitable format also that the contents have been included in an appendix	2 days	Mon 01/05/17	Tue 02/05/17	7,14,24,21
Submit to printer and binders and prepare digital copies	8 days	Wed 03/05/17	Fri 12/05/17	

APPENDIX C – INTERIM PROGRESS REPORT

Department of Computing and Informatics

Undergraduate Project Interim Review

To be completed and signed by the Supervisor and student during week **commencing 6 March 2017.**

Student: RUSSELL CLARKE [redacted]	**Supervisor:** MARCIN BUDKA

Assessment

1. Define the problem *Has the problem been defined, has the artefact been identified and have objectives been set?*	Choose an item. YES
Comments:	
2. Review other work *Is there evidence of appropriate research?*	Choose an item. YES
Comments:	
3. Analysis, Design & Implementation of artefact *Is there evidence of appropriate analysis of the problem and development of a solution?*	Choose an item. TSE
Comments:	
4. Dissertation *Have sections of the dissertation been written and has the Supervisor seen these?*	Choose an item. YES
Comments:	
5. Planning & progress *Is there an acceptable plan for this project and is it being followed?*	Choose an item. TSE
Comments: Slightly behind schedule	
6. Overall assessment	Choose an item. SATISFACTORY / UNCERTAIN

Signed:

Supervisor: Bueller ..... Student: CLARKE .....

Date: 09/03/2017

Supervisor to retain the signed form and supply the student with a copy if required.

Supervisor to upload the form on MyBU and grade the student as 1 (satisfactory), 0.5 (uncertain) and 0 (unsatisfactory).

Supervisor to notify the Project Tutor if the student is at risk of failing the Project.

APPENDIX D – TECHNICAL DOCUMENTATION AND DESIGNS

D.1 SYSTEM REQUIREMENT AND SPECIFICATIONS

HP-15-TS-Notebook-PC

Linux 4.4.0-66-generic #87-Ubuntu SMP x86_64 GNU/Linux as of Fri Mar 3 15:29:05 UTC 2017

Eclipse Release 4.6.0 (Neon) Last revised 30 May 2016 – Available from: http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/neon/3/eclipse-jee-neon-3-linux-gtk-x86_64.tar.gz and then upgraded to 4.6.0 from within the IDE.

PyDev 5.5.0.201701191708

OpenAI Gym image February 2016 release: https://travis-ci.org/OpenAI/gym.svg?branch=master (Author retains a copy)

Results Tabularised and Graphed LibreOffice Version: 5.3.1.2 Build ID: 1:5.3.1-0ubuntu1~xenial0 Locale: en-GB (en_GB.UTF-8)

Microsoft Office 2016 and Microsoft Visio and Project 2013 (all at latest revisions / updates)

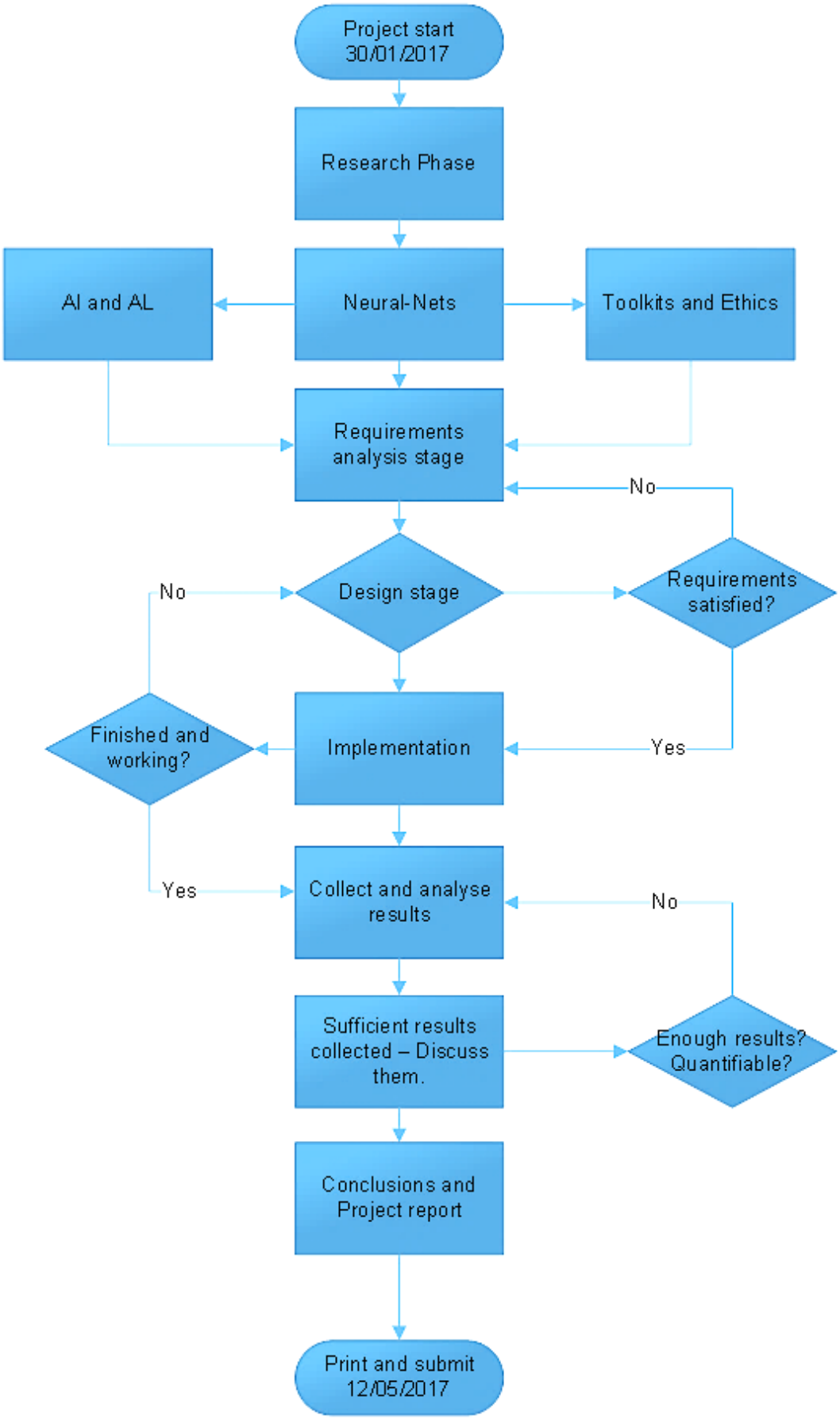
Python 2.7.12
[Installed packages under Python, most of which are installed when gym or Python is installed]

atari-py (0.0.18)	imageio (2.1.2)	numpy (1.12.0)	pyzmq (16.0.2)
backports-abc (0.5)	ipykernel (4.5.2)	pachi-py (0.0.21)	qtconsole (4.2.1)
backports.shutil-get-terminal-size (1.0.0)	ipython (5.3.0)	pandocfilters (1.4.1)	requests (2.13.0)
beautifulsoup4 (4.4.1)	ipython-genutils (0.1.0)	pathlib2 (2.2.1)	scandir (1.5)
bleach (2.0.0)	ipywidgets (6.0.0)	pexpect (4.0.1)	scipy (0.18.1)
boto (2.38.0)	Jinja2 (2.9.5)	pickleshare (0.7.4)	setuptools (20.7.0)
box2d-py (2.3.1)	jsonschema (2.6.0)	Pillow (3.1.2)	simplegeneric (0.8.1)
certifi (2017.1.23)	jupyter (1.0.0)	pip (9.0.1)	singledispatch (3.4.0.3)
chardet (2.3.0)	jupyter-client (5.0.0)	prompt-toolkit (1.0.13)	six (1.10.0)
configobj (5.0.6)	jupyter-console (5.1.0)	ptyprocess (0.5)	terminado (0.6)
configparser (3.5.0)	jupyter-core (4.3.0)	pycrypto (2.6.1)	testpath (0.3)
decorator (4.0.6)	Keras (1.2.2)	pyglet (1.2.4)	Theano (0.8.2)
deja-dup-caja (0.0.4)	lockfile (0.12.2)	Pygments (2.2.0)	tornado (4.4.2)
duplicity (0.7.6)	lxml (3.5.0)	pygobject (3.20.0)	traitlets (4.3.2)
entrypoints (0.2.2)	MarkupSafe (1.0)	pygpgme (0.3)	urllib3 (1.13.1)
enum34 (1.1.6)	mate-menu (5.7.1)	PyOpenGL (3.1.0)	vboxapi (1.0)
folder-color-caja (0.0.79)	mistune (0.7.3)	Pyste (0.9.10)	virtualenv (15.0.1)
folder-color-common (0.0.79)	mujoco-py (0.5.7)	python-cloudfiles (1.7.10)	wcwidth (0.1.7)
functools32 (3.2.3.post2)	nbconvert (5.1.1)	python-dateutil (2.6.0)	webencodings (0.5)
gym (0.7.4.dev0, /home/russia/gym)	nbformat (4.3.0)	python-xlib (0.14)	wheel (0.29.0)
html5lib (0.999999999)	netifaces (0.10.4)	pyxdg (0.25)	widgetsnbextension (2.0.0)
httplib2 (0.9.1)	notebook (4.4.1)	PyYAML (3.12)	zenmap (7.1)

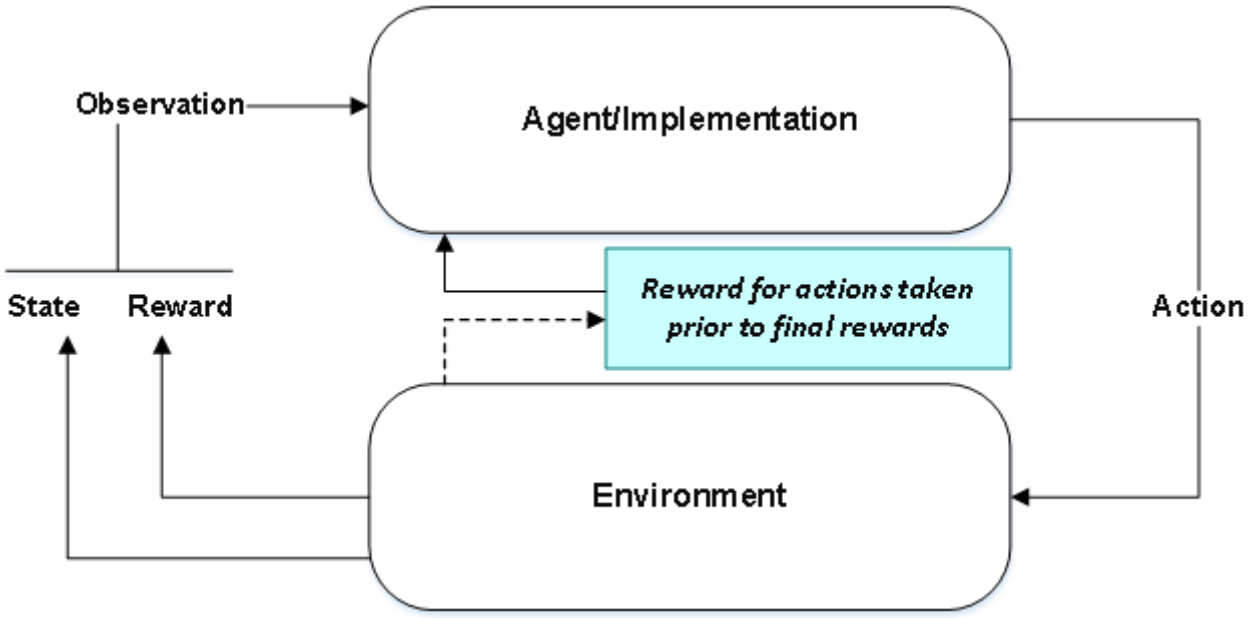
Ensure all Operating system components are up to date, Install the Eclipse IDE and if required ensure you have updated java and have java SDK installed, version 7 was installed in this environment. Install Python 2.x (2017a) and any relevant scripts using pip. Proceed to install gym and all its components (OpenAI 2016c), install any required gym environment dependencies. From within the IDE install PyDev for python development.

D.2 Project and Software Designs

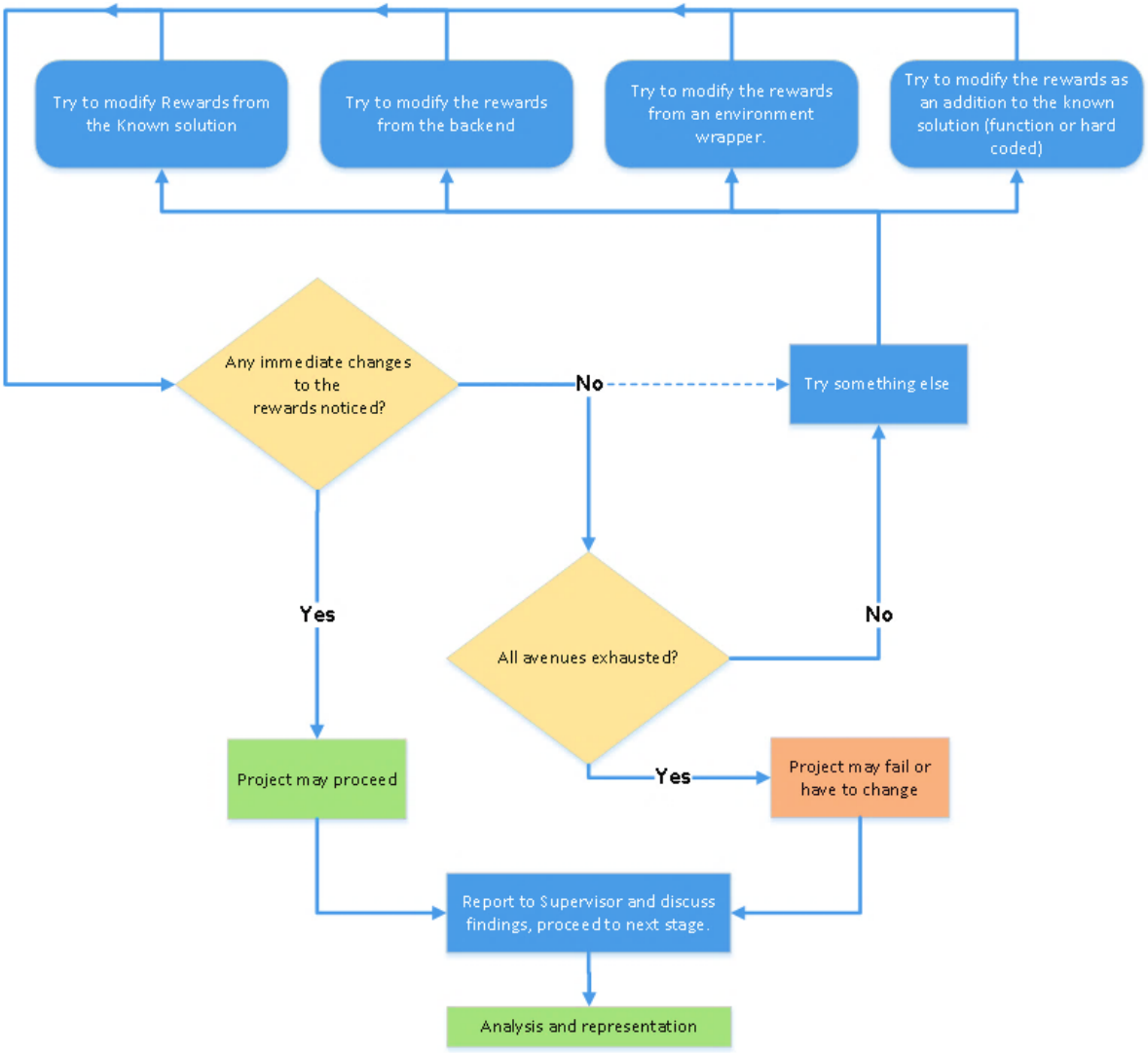
D.2.1 Project flow



D.2.2 Software Design – RL state transition model with immediate rewards



D.2.3 Reward value modification process and plan

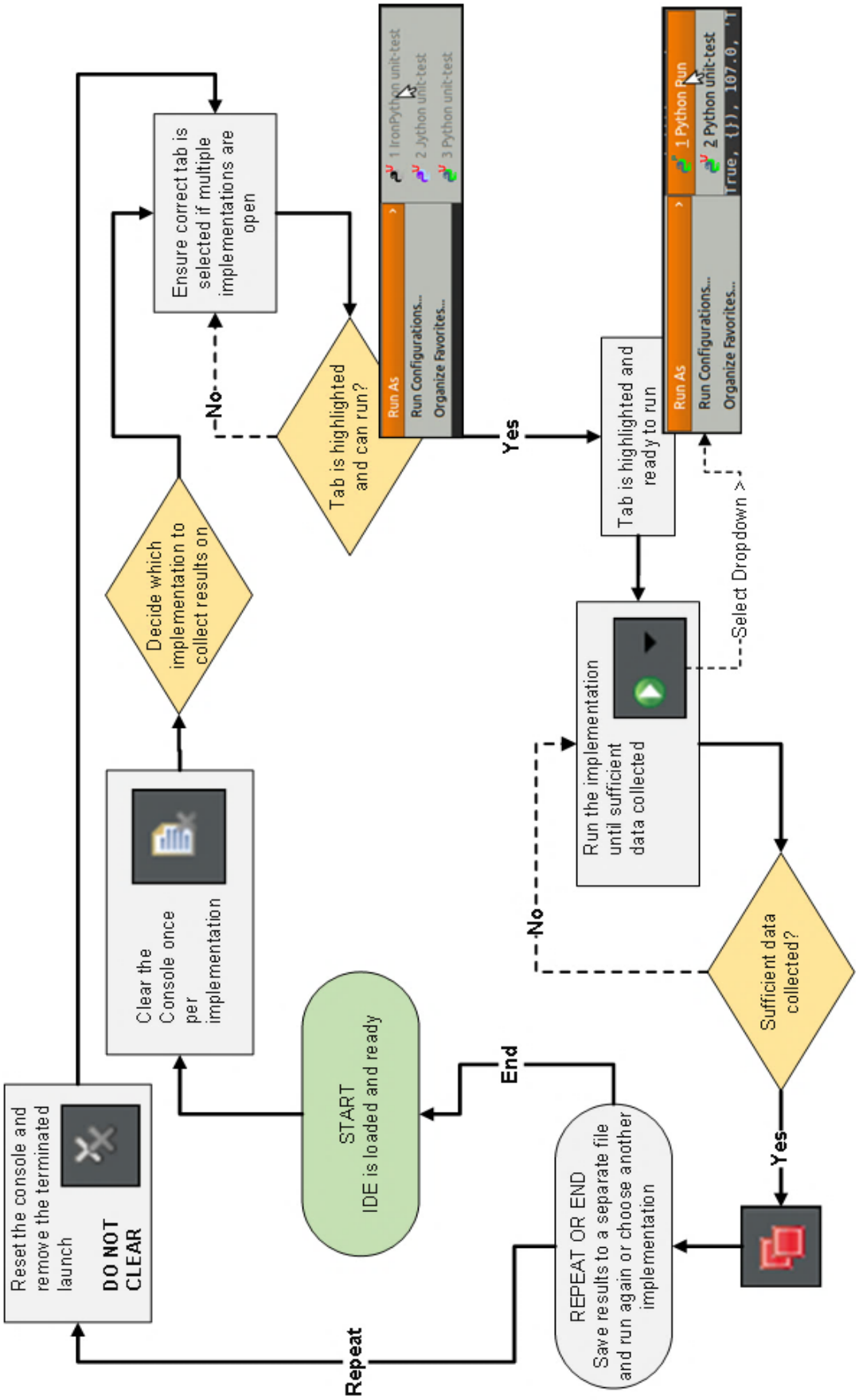


D.2.4 Detailed overview of methodologies and processes/life-cycles considered

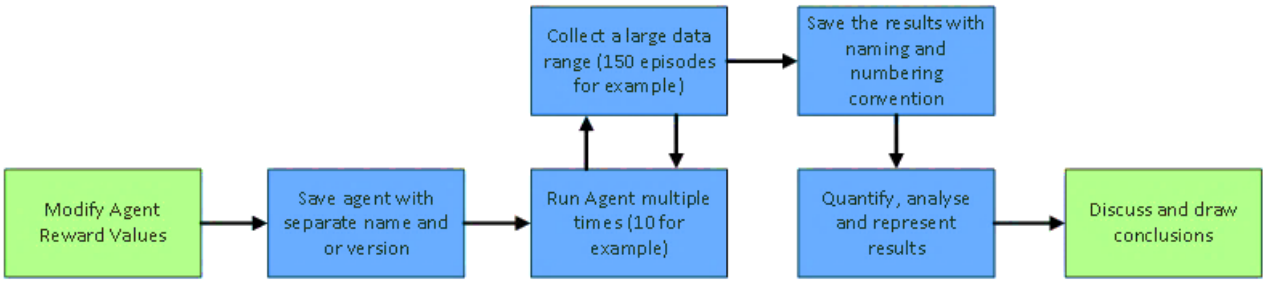
Methodology Type	Characteristics	Methods	Justification
Empirical	• Verifiable • Cumulative • Exploration • Trial and error • Resulting product finite	• Identify and compose research question • Selecting subjects to use to answer • How to use subjects to answer • Analysing results • Interpretation and communication of results	• Goal-orientated • Verifiable in nature • Subjective study resources • Holds research design methods • Quantifiable approach • Logical structure • Lesser philosophical approach
Qualitative	• Ethical approach • Systematic • Rigorous • Strategic • Data generation methods • Contextual • Flexible • Critical self-scrutiny • Reactive • Analytical • Generating Hypothesis	• Planning and designing the research • Research reasoning • Strategic research • Consider correlation of research subject and data type generated • Generating Data • Sampling and selecting data • Sorting, Organising data • Analysis and production of convincing explanations • Conclusive	• Research design, not always possible before research is started • Translates well to other disciplines. • Analytical and conclusive • Ethical approach in generation and collection of data. • A flexible and contextual rigorous systematic strategy • Self-scrutiny of approach and reasoning
Quantitative	• Control • Operational Definition • Replication • Hypothesis Testing	• Generating and testing theory • Controlled sampling and design • Quantifiable qualitative data collection • Statistical data collection • Documented methods for replication	• Precision Quantitative and reliable measurement • Controlled sampling and design • Controlled experimentation Statistical analyses • Replicable
Incremental	• Unidirectional • Distinct set of methods • Quality assurance • Incremental • Finite	• Requirements analysis • Risk assessment • Design • Programming • Testing • Maintenance	• Overly Rigid though replicable • Suited to variety of designs • Throwaway prototypes • High risk as lack of functional product to review • Risk of software degradation due to continual improvement
Evolutionary	• Iterative • Incremental • Bidirectional • Distinct set of methods • Prototype and improvement strategy • Evolution	• Requirements analysis • Design • Program • Test • Verify • Repeat	• Risk of software degradation due to continual improvement • Product is never truly complete • Multiple functional prototypes • Continual end-user feedback
Hybrid/Dynamic	• Combination of many approaches • Suited to all or most tasks • Adaptable • Reactive	• Combine any of the above methodologies, methods or approaches to suit the task in hand	• All the advantages and disadvantages of the methodologies, methods and approaches above

D.3 Test Plans and Result gathering

D.3.1 Result collection process



D.3.2 Result collection flow per agent



D.3.3 Reward values to be tested

Reward Type	*Neutral*	*Negative*	*Positive*
High	200	-200	200
Medium	100	-100	100
Low	10	-10	10
Original	0	-Infinite or -1	Infinite or 1
High	-200	200	-200
Medium	-100	100	-100
Low	-10	10	-10

D.3.4 Data generation and sample collection

Developer	**Agent No. / Developer Solution**	**Trials / Agent No.**	**Episode count (sample) / Trial**
Kevin Frans (CartPole)	1 (Original solution, no reward modifications)	10	First 150
Andreij Karpathy (Atari Pong)	2 (Effects of Null reward modifications)	10	First 150
	3 (Effects of negative reward modifications)	10	First 150
Dhruv Parthasarathy (Atari Pong)	4 (Effects of positive reward modifications)	10	First 150

D.4.1 Frequencies of reward occurrences per Agent

Reward Frequencies

Kevin Frans

Reward	frequencies			
Original	200 Frequency	75	Higher than or equal to 135	161
Zero	200 Frequency	73	Higher than or equal to 135	153
Negative	200 Frequency	508	Higher than or equal to 135	657
Positive	200 Frequency	18	Higher than or equal to 135	80

Andrej Karpathy

Reward Value	0	1	2	3	4	5	6
Original	928	398	139	41	12	2	1
Zero	889	427	146	38	7	3	0
Negative	866	422	166	49	5	1	2
Positive	891	404	170	31	13	1	0

Dhruv Parthasarathy

Reward Value	0	1	2	3	4	5	6
Original	890	433	134	40	13	0	0
Zero	907	419	136	35	10	3	0
Negative	918	404	139	44	3	1	0
Positive	929	383	134	54	8	2	0

D.4.2 Frans results from four agents

Original		Ob1 Pole	Ob2 Cart	Ob3 Pole	Ob4 Cart	Rewards	Time 150 episodes		Ob1 Pole	Ob2 Cart	Ob3 Pole	Ob4 Cart	Rewards	Time 150 episodes
Set1	Mean Average	-0.0027	-0.006413333333333	-0.048613333333333	-0.154046666666667	43.3733333333333	1.34184	Total	-0.405000000000001	-0.962000000000002	-7.292	-23.107	6506	201.276
Set2	Mean Average	0.008126666666667	0.055993333333333	-0.006406666666667	-0.076226666666667	9.81333333333333	0.30036	Total	1.219	8.399	-0.961	-11.434	1472	45.054
Set3	Mean Average	0.010484	0.141432666666667	-0.016710666666667	-0.2163	9.6	0.306855533333333	Total	1.5726	21.2149	-2.5066	-32.445	1440	46.02833
Set4	Mean Average	-0.015376	-0.044264666666667	-0.013142666666667	-0.016856	83.5466666666667	2.14026533333333	Total	-2.321776	-6.6397	-1.9714	-2.5284	12532	321.0398
Set5	Mean Average	0.018880666666667	-0.103866	-0.060480666666667	-0.000737333333333	156.993333333333	4.47143733333333	Total	2.8321	-15.5799	-9.0721	-0.110600000000001	23549	670.7156
Set6	Mean Average	-0.055268	-0.061278	-0.002562	0.003491333333333	120.726666666667	3.00011	Total	-8.345468	-9.252978	-0.386862000000001	0.527191333333333	18229.7266666667	450.0165
Set7	Mean Average	-0.045328666666667	-0.116942666666667	0.024458	0.133174	72.62	1.8258	Total	-6.7993	-17.5414	3.6687	19.9761	10893	273.87
Set8	Mean Average	0.001976666666667	0.056143333333333	-0.007204666666667	-0.091368666666667	9.69333333333333	0.337303	Total	0.2965	8.4215	-1.0807	-13.7053	1454	21.43855
Set9	Mean Average	-0.004319333333333	-0.010712	0.000680666666667	0.018428	9.42	0.342391333333333	Total	-0.6479	-1.6068	0.1021	2.7642	1413	4.4933
Set10	Mean Average	-0.003007333333333	-0.019208666666667	0.005294	0.041120666666667	9.12	0.311710066666667	Total	-0.4511	-2.8813	0.794099999999999	6.16810000000001	1368	24.73551
Mean Average	Original Deviance	0.022733738150941	0.074701294644857	0.023758437452923	0.095826037121349	51.2429730423813	1.36194255914697	Original Deviance	3.42186846052513	11.2093401011644	3.56365886887311	14.3740492414123	7702.59049465335	212.693797913468
	Original Mean	-0.008653133333333	-0.0109116	-0.0124688	-0.035932133333333	52.4906666666667	1.43780726	Original Mean	-1.3050344	-1.6428678	-1.8705762	-5.38947086666667	7885.67266666667	205.886759
	Original Mean Total	-0.086531333333333	-0.109116	-0.124688	-0.359321333333333	524.906666666667	14.3780726	Original Total	-13.050344	-16.428678	-18.705762	-53.8947086666667	78856.7266666667	2058.66759
	Original Deviation %	-26.2722614747734	-68.4604408563689	-19.0543095188974	-26.6686189301357	9.76230181410943	9.47235834062325	Original Deviation %	-26.2205230798907	-68.2303232260344	-19.0511291059574	-26.6706131214371	9.76782935362354	10.3316241508163

High zero		Ob1 Pole	Ob2 Cart	Ob3 Pole	Ob4 Cart	Rewards	Time 150 episodes		Ob1 Pole	Ob2 Cart	Ob3 Pole	Ob4 Cart	Rewards	Time 150 episodes
Set1	Mean Average	-0.01004	-0.101680666666667	0.011104666666667	0.150624666666667	8.96666666666667	0.315632733333333	Total	-1.506	-15.2521	1.6657	22.5937	1345	47.34491
Set2	Mean Average	-0.009872	0.046161333333333	-0.06505	-0.033588	112.693333333333	3.587792	Total	-1.4808	6.9242	-9.7575	-5.0382	16904	538.1688
Set3	Mean Average	0.001449333333333	0.009777333333333	0.001098666666667	-0.007620666666667	9.42	0.330784933333333	Total	0.2174	1.4666	0.164800000000001	-1.1431	1413	49.61774
Set4	Mean Average	-0.06951	-0.154850666666667	-0.044672	-0.044957333333333	151.2	4.68277466666667	Total	-10.4265	-23.2276	-6.7008	-6.7436	22680	702.4162
Set5	Mean Average	0.014873333333333	0.035116666666667	0.012923333333333	0.028380666666667	82.2933333333333	2.62154266666667	Total	2.231	5.2675	1.9385	4.2571	12344	393.2314
Set6	Mean Average	-0.008146666666667	-0.045039333333333	0.019175333333333	0.108408	23.32	0.759502666666667	Total	-1.222	-6.7559	2.8763	16.2612	3498	113.9254
Set7	Mean Average	0.001714	0.002405333333333	-0.001883333333333	-0.012606666666667	9.21333333333333	0.318015333333333	Total	0.2571	0.3608	-0.2825	-1.891	1382	47.7023
Set8	Mean Average	-0.003024666666667	-0.03104	0.003506666666667	0.052388	9.63333333333333	0.330423	Total	-0.4537	-4.656	0.526	7.85820000000001	1445	49.56345
Set9	Mean Average	-0.009191333333333	-0.177084666666667	0.008496	0.245731333333333	16.2533333333333	0.5405956	Total	-1.3787	-26.5627	1.2744	36.8597	2438	81.08934
Set10	Mean Average	-0.014618	-0.066880666666667	-0.024850666666667	-0.042216	63.6733333333333	0.013649555555556	Total	-2.1927	-10.0321	-3.7276	-6.3324	9551	307.124
Mean Average	Zero Deviance	0.021209580888415	0.072984314169888	0.026361855793804	0.09152076795482	48.84474014671 39	1.572730725414	Zero Deviance	3.18143713326226	10.9476471254832	3.95427836907065	13.728115193223	7326.71102200708	227.592774727747
	Zero Mean	-0.0106366	-0.048311533333333	-0.008015133333333	0.0444544	48.6666666666667	1.35007135555556	Zero Mean	-1.59549	-7.24673	-1.20227	6.66816	7300	233.018354
	Zero Mean Total	-0.106366	-0.483115333333333	-0.080151333333333	0.444544	486.666666666667	13.5007135555556	Zero Total	-15.9549	-72.4673	-12.0227	66.6816	73000	2330.18354
	Zero Deviation %	-19.9401884891931	-15.107016717172	-32.8901026314442	20.5875611761 31	10.0365904411056	11.6492414933 84	Zero Deviation %	-19.940188489193	-15.107016717172	-32.8901026314443	20.5875611761 31	10.0365904411056	9.76716086183268

High Negative		Ob1 Pole	Ob2 Cart	Ob3 Pole	Ob4 Cart	Rewards	Time 150 episodes		Ob1 Pole	Ob2 Cart	Ob3 Pole	Ob4 Cart	Rewards	Time 150 episodes
Set1	Mean Average	0.032701333333333	-0.071696666666667	-0.032985333333333	-0.056026	145.566666666667	4.62581733333333	Total	4.9052	-10.7545	-4.9478	-8.4039	21835	693.8726
Set2	Mean Average	0.08486	0.001365333333333	-0.011228666666667	0.027098	171.74	5.43862666666667	Total	12.729	0.204799999999999	-1.6843	4.0647	25761	815.794
Set3	Mean Average	0.026265333333333	0.363924	-0.045934666666667	-0.569635333333333	9.24	0.3195572	Total	3.9398	54.5886	-6.8902	-85.4453	1386	47.93358
Set4	Mean Average	0.075778	0.076411333333333	-0.074565333333333	0.017291333333333	89.84	2.87072266666667	Total	11.3667	11.4617	-11.1848	2.5837	13476	430.6084
Set5	Mean Average	0.025077333333333	0.020626	-0.050866	0.072594666666667	196.406666666667	6.20792933333333	Total	3.7616	3.0939	-7.6299	10.8892	29461	931.1894
Set6	Mean Average	0.021459333333333	0.257292666666667	-0.032707333333333	-0.414390666666667	9.11333333333333	0.318807866666667	Total	3.2189	38.5939	-4.9061	-62.1586	1367	47.82118
Set7	Mean Average	-0.001579333333333	-0.024484666666667	0.002724	0.033509333333333	8.93333333333333	0.313540333333333	Total	-0.2369	-3.6727	0.4086	5.02640000000001	1340	47.03105
Set8	Mean Average	-0.013526666666667	-0.053704666666667	-0.029993333333333	-0.068532	54.6866666666667	1.757158	Total	-2.029	-8.0557	-4.499	-10.2798	8203	263.5737
Set9	Mean Average	-0.055583333333333	-0.072761333333333	0.038232666666667	0.069152666666667	176.133333333333	5.74473133333333	Total	-8.3375	-10.9142	5.7349	10.3729	26420	861.7097
Set10	Mean Average	-0.067995333333333	-0.058060666666667	-0.005730666666667	-0.019146666666667	201.126666666667	6.36102666666667	Total	-10.1993	-8.7091	-0.8596	-2.872	30169	954.154
Mean Average	Negative Deviance	0.046974421307535	0.142453929708302	0.030058705526649	0.208363300063039	76.8721601274913	2.43508790860893	Negative Deviance	7.04616319613022	21.3680894562453	4.5088058289973	31.2544950094558	11530.8240191237	365.26318629134
	Negative Mean	0.012745666666667	0.043891133333333	-0.024305466666667	-0.090808466666667	106.278666666667	3.39579174	Negative Mean	1.91185	6.58367	-3.64582	-13.62127	15941.8	509.368761
	Negative Mean Total	0.127456666666667	0.438911333333333	-0.243054666666667	-0.908084666666666	1062.78666666667	33.9579174	Negative Total	19.1185	65.8367	-36.4582	-136.2127	159418	5093.68761
	Negative Deviation %	36.855209331957	32.4561976165958	-12.3670555019099	-22.9453604615839	7.23307532344 13	7.17089885084923	Negative Deviation %	36.8552093319571	32.4561976165958	-12.3670555019099	-22.9453604615838	7.23307532344133	7.17089885084923

High Positive		Ob1 Pole	Ob2 Cart	Ob3 Pole	Ob4 Cart	Rewards	Time 150 episodes		Ob1 Pole	Ob2 Cart	Ob3 Pole	Ob4 Cart	Rewards	Time 150 episodes
Set1	Mean Average	-0.008151333333333	-0.079832	0.009732666666667	0.129406	19.6	0.6451836	Total	-1.2227	-11.9748	1.4599	19.4109	2940	96.77754
Set2	Mean Average	0.012145333333333	0.114823333333333	-0.014795333333333	-0.184318666666667	9.54	0.3316066	Total	1.8218	17.2235	-2.2193	-27.6478	1431	49.74099
Set3	Mean Average	0.008870666666667	0.083643333333333	-0.011387333333333	-0.13093	9.71333333333333	0.345591133333333	Total	1.3306	12.5465	-1.7081	-19.6395	1457	51.83867
Set4	Mean Average	-0.008986666666667	-0.083046	0.013417333333333	0.143958666666667	9.50666666666667	0.332466	Total	-1.348	-12.4569	2.0126	21.5938	1426	49.8699
Set5	Mean Average	0.010798	0.156277333333333	-0.019206	-0.246758666666667	9.04666666666667	0.323976933333333	Total	1.6197	23.4416	-2.8809	-37.0138	1357	48.59654
Set6	Mean Average	-0.005367333333333	-0.053244	0.005718	0.087695333333333	9.63333333333333	0.335170933333333	Total	-0.8051	-7.9866	0.8577	13.1543	1445	50.27564
Set7	Mean Average	0.089721333333333	0.109589333333333	0.020824666666667	0.016425333333333	135.266666666667	4.27548133333333	Total	13.4582	16.4384	3.1237	2.4638	20290	641.3222
Set8	Mean Average	-0.033448	-0.341614666666667	0.046883333333333	0.546898666666667	9.40666666666667	0.334094133333333	Total	-5.0172	-51.2422	7.0325	82.0348	1411	50.11412
Set9	Mean Average	0.004044666666667	0.086712666666667	-0.002498	-0.000443333333333	70.4933333333333	2.25385333333333	Total	0.606700000000002	13.0069	-0.3747	-0.066499999999999	10574	338.078
Set10	Mean Average	-0.013223333333333	-0.146568	0.016509333333333	0.225060666666667	9.44	0.333992133333333	Total	-1.9835	-21.9852	2.4764	33.7591	1416	50.09882
Mean Average	Positive Deviance	0.030920181053444	0.146993457553215	0.018735831998962	0.217423688954988	39.7149125140671	1.24537358748784	Positive Deviance	4.63802715801665	22.0490186329823	2.81037479984432	32.6135533432483	5957.23687711006	186.806038123175
	Positive Mean	0.005640333333333	-0.015325866666667	0.006519866666667	0.0586994	29.1646666666667	0.951141613333333	Positive Mean	0.84605	-2.29888	0.97798	8.80491	4374.7	142.671242
	Positive Mean Total	0.056403333333333	-0.153258666666667	0.065198666666667	0.586994	291.646666666667	9.51141613333333	Positive Total	8.4605	-22.9888	9.7798	88.0491	43747	1426.71242
	Positive Deviation %	54.8197761127196	-95.9120033798293	28.7365263077396	37.0401893298719	13.6174752031226	13.0934612683315	Positive Deviation %	54.8197761127197	-95.9120033798297	28.7365263077396	37.0401893298 72	13.6174752031226	13.0934612683315

D.4.3 Karpathy results from four agents

Original		EpisodeTime	AvgTime/Game	Reward Total	Running/mean	Actual rewards		EpisodeTime	AvgTime/Game	Reward Total	Running/mean	Actual rewards
Set1	Mean Average	20.3055873333333	0.944465333333333	-20.6533333333333	-19.8764473333333	0.486666666666667	Total	50.7639683333333	2.36116333333333	-3098	-2981.4671	73
Set2	Mean Average	19.608938	0.904384	-20.4866666666667	-20.2882966666667	0.653333333333333	Total	49.022345	2.26096	-3073	-3043.2445	98
Set3	Mean Average	19.7127853333333	0.914945333333333	-20.6133333333333	-19.8489033333333	0.526666666666667	Total	49.2819633333333	2.28736333333333	-3092	-2977.3355	79
Set4	Mean Average	19.5896366666667	0.910506666666667	-20.66	-20.90652	0.48	Total	48.9740916666667	2.27737666666667	-3099	-3135.978	72
Set5	Mean Average	19.209884	0.891694666666667	-20.6133333333333	-20.891514	0.526666666666667	Total	48.02471	2.22923666666667	-3092	-3133.7271	79
Set6	Mean Average	19.0659353333333	0.882046666666667	-20.54	-20.852826	0.6	Total	47.6648383333333	2.20511666666667	-3081	-3127.9239	90
Set7	Mean Average	19.901582	0.920051333333333	-20.5333333333333	-19.833738	0.606666666666667	Total	49.753955	2.30012833333333	-3080	-2975.0607	91
Set8	Mean Average	19.3423126666667	0.895201333333333	-20.5466666666667	-19.8424006666667	0.593333333333333	Total	48.3557816666667	2.23800333333333	-3082	-2976.3601	89
Set9	Mean Average	18.7718613333333	0.871761333333334	-20.64	-20.88495	0.5	Total	46.9296533333333	2.17940333333333	-3096	-3132.7425	75
Set10	Mean Average	19.2636573333333	0.891079333333333	-20.54	-19.2978973333333	0.6	Total	48.1591433333333	2.22769833333333	-3081	-2894.6846	90
Mean Average	Original Deviance	0.41877406491191	0.019905778738849	0.057209167874466	0.561859567882856	0.057209167874467	Original Deviance	1.04693516227978	0.049764446847121	8.58137518116997	84.2789351824283	8.58137518116997
	Original Mean	19.477218	0.902658	-20.5826666666667	-20.2523493333333	0.557333333333333	Original Mean	48.693045	2.256645	-3087.4	-3037.8524	83.6
	Original Mean Total	194.77218	9.02658	-205.826666666667	-202.523493333333	5.57333333333333	Original Total	486.93045	22.56645	-30874	-30378.524	836
	Original Deviation %	0.215007125202331	0.220524038327346	-0.027794827949634	-0.27742932863502	1.02648028482894	Original Deviation %	0.215007125202331	0.220524038327346	-0.027794827949634	-0.27742932863502	1.02648028482894

High Zero		EpisodeTime	AvgTime/Game	Reward Total	Running/mean	Actual rewards		EpisodeTime	AvgTime/Game	Reward Total	Running/mean	Actual rewards
Set1	Mean Average	19.2648086666667	0.894104666666667	-20.6066666666667	-20.868126	0.533333333333333	Total	48.1620216666667	2.23526166666667	-3091	-3130.2189	80
Set2	Mean Average	19.2204353333333	0.893722	-20.66	-20.3761233333333	0.48	Total	48.0510883333333	2.234305	-3099	-3056.4185	72
Set3	Mean Average	19.4133166666667	0.899686	-20.58	-20.890832	0.56	Total	48.5332916666667	2.249215	-3087	-3133.6248	84
Set4	Mean Average	19.5908486666667	0.909130666666667	-20.6	-20.3615713333333	0.54	Total	48.9771216666667	2.27282666666667	-3090	-3054.2357	81
Set5	Mean Average	19.7492226666667	0.910371333333333	-20.4733333333333	-20.30832	0.666666666666667	Total	49.3730566666667	2.27592833333333	-3071	-3046.248	100
Set6	Mean Average	19.172758	0.886054	-20.5333333333333	-20.8417586666667	0.606666666666667	Total	47.931895	2.215135	-3080	-3126.2638	91
Set7	Mean Average	19.5810513333333	0.90406	-20.5066666666667	-19.79826	0.633333333333333	Total	48.9526283333333	2.26015	-3076	-2969.739	95
Set8	Mean Average	19.2744353333333	0.891135333333333	-20.5333333333333	-20.8529166666667	0.606666666666667	Total	48.1860883333333	2.22783833333333	-3080	-3127.9375	91
Set9	Mean Average	19.8159286666667	0.913224666666667	-20.4666666666667	-19.7389726666667	0.673333333333333	Total	49.539821666666667	2.28306166666667	-3070	-2960.8459	101
Set10	Mean Average	19.6862686666667	0.912541333333333	-20.6	-20.8808433333333	0.54	Total	49.2156716666667	2.28135333333333	-3090	-3132.1285	81
Mean Average	Zero Deviance	0.224735645842043	0.009312626135164	0.060088823142634	0.426243011773768	0.060088823142633	Zero Deviance	0.561839114605108	0.023281565337909	9.013323471395	63.9364517660653	9.013323471395
	Zero Mean	19.4769074	0.901403	-20.556	-20.4917724	0.584	Zero Mean	48.6922685	2.2535075	-3083.4	-3073.76586	87.6
	Zero Mean Total	194.769074	9.01403	-205.56	-204.917724	5.84	Zero Total	486.922685	22.535075	-30834	-30737.6586	876
	Zero Deviation %	0.115385693029502	0.103312570905174	-0.029231768409532	-0.20800690318704	1.02891820449715	Zero Deviation %	0.115385693029502	0.103312570905175	-0.029231768409532	-0.20800690318704	1.02891820449715

High Negative		EpisodeTime	AvgTime/Game	Reward Total	Running/mean	Actual rewards		EpisodeTime	AvgTime/Game	Reward Total	Running/mean	Actual rewards
Set1	Mean Average	19.2867046666667	0.893924	-20.58	-20.3412913333333	0.56	Total	48.2167616666667	2.23481	-3087	-3051.1937	84
Set2	Mean Average	19.2565353333333	0.890561333333333	-20.54	-20.8454106666667	0.6	Total	48.1413383333334	2.22640333333333	-3081	-3126.8116	90
Set3	Mean Average	19.1831513333333	0.883484	-20.46	-20.2748613333333	0.68	Total	47.9578783333333	2.20871	-3069	-3041.2292	102
Set4	Mean Average	19.9721106666667	0.919522	-20.4666666666667	-20.332854	0.693333333333333	Total	49.9302766666667	2.298805	-3067	-3049.9281	104
Set5	Mean Average	19.6765753333333	0.904854	-20.4133333333333	-20.7771633333333	0.726666666666667	Total	49.1914383333333	2.262135	-3062	-3116.5745	109
Set6	Mean Average	19.1134893333333	0.885542	-20.5733333333333	-20.8801206666667	0.566666666666667	Total	47.7837233333333	2.213855	-3086	-3132.0181	85
Set7	Mean Average	19.405788	0.892928666666667	-20.4333333333333	-20.8132926666667	0.706666666666667	Total	48.51447	2.23232166666667	-3065	-3121.9939	106
Set8	Mean Average	20.3271233333333	0.942562666666667	-20.5933333333333	-20.8616506666667	0.546666666666667	Total	50.8178083333334	2.35640666666667	-3089	-3129.2476	82
Set9	Mean Average	20.1326813333333	0.93152	-20.54	-19.8320626666667	0.6	Total	50.3317033333333	2.3288	-3081	-2974.8094	90
Set10	Mean Average	19.5235413333333	0.904392666666667	-20.5733333333333	-20.876052	0.566666666666667	Total	48.8088533333333	2.26098166666667	-3086	-3131.4078	85
Mean Average	Negative Deviance	0.403166686800877	0.019089040676786	0.065662436411967	0.345144894476436	0.065662436411967	Negative Deviance	1.00791671700219	0.047722601691966	9.84936546179499	51.7717341714654	9.84936546179499
	Negative Mean	19.5877700666667	0.904929133333333	-20.5153333333333	-20.5834759333333	0.624666666666667	Negative Mean	48.9694251666667	2.26232283333333	-3077.3	-3087.52139	93.7
	Negative Mean Total	195.877700666667	9.04929133333333	-205.153333333333	-205.834759333333	6.24666666666667	Negative Total	489.694251666667	22.6232283333333	-30773	-30875.2139	937
	Negative Deviation %	0.205825719532497	0.210945144471934	-0.032006516952507	-0.167680568429893	1.05115960104536	Negative Deviation %	0.205825719532496	0.210945144471934	-0.032006516952507	-0.167680568429893	1.05115960104536

High Positive		EpisodeTime	AvgTime/Game	Reward Total	Running/mean	Actual rewards		EpisodeTime	AvgTime/Game	Reward Total	Running/mean	Actual rewards
Set1	Mean Average	19.806126	0.916564666666667	-20.5533333333333	-20.870056	0.586666666666667	Total	49.515315	2.29141166666667	-3083	-3130.5084	88
Set2	Mean Average	19.060424	0.884556666666667	-20.6133333333333	-20.8764706666667	0.526666666666667	Total	47.65106	2.21139166666667	-3092	-3131.4706	79
Set3	Mean Average	19.325724	0.894176666666667	-20.5666666666667	-20.8594413333333	0.573333333333333	Total	48.31431	2.23544166666667	-3085	-3128.9162	86
Set4	Mean Average	19.0708186666667	0.886814	-20.6533333333333	-20.9098826666667	0.486666666666667	Total	47.6770466666667	2.217035	-3098	-3136.4824	73
Set5	Mean Average	18.9898286666667	0.879458	-20.5733333333333	-20.8770633333333	0.566666666666667	Total	47.4745716666667	2.198645	-3086	-3131.5595	85
Set6	Mean Average	19.524918	0.89663	-20.38	-20.7799766666667	0.76	Total	129.226488757569	5.97702633878556	-3057	-3116.9965	114
Set7	Mean Average	19.4431446666667	0.898370666666667	-20.5333333333333	-20.823938	0.606666666666667	Total	48.6078616666667	6.05438900316369	-3080	-3123.5907	91
Set8	Mean Average	19.181286	0.890854666666666	-20.62	-20.3742533333333	0.52	Total	132.935516638689	2.22713666666667	-3093	-3056.138	78
Set9	Mean Average	19.366192	0.892868	-20.48	-20.8224086666667	0.66	Total	48.41548	2.23217	-3072	-3123.3613	99
Set10	Mean Average	19.801302	0.912358666666667	-20.4666666666667	-20.7976346666667	0.673333333333333	Total	49.503255	2.28089666666667	-3070	-3119.6452	101
Mean Average	Positive Deviance	0.277731118532246	0.011044038217971	0.078156538078682	0.146641020507694	0.078156538078682	Positive Deviance	33.0915798359835	1.51192486178453	11.7234807118023	21.996153076154	11.7234807118023
	Positive Mean	19.3569764	0.8952652	-20.544	-20.7991125333333	0.596	Positive Mean	64.9320905396258	2.99255436752826	-3081.6	-3119.86688	89.4
	Positive Mean Total	193.569764	8.952652	-205.44	-207.991125333333	5.96	Positive Total	649.320905396258	29.9255436752826	-30816	-31198.6688	894
	Positive Deviation %	0.14347856441683	0.123360521753453	-0.038043486214312	-0.07050349877798	1.311351310045	Positive Deviation %	5.09633673596092	5.05228870088441	-0.038043486214312	-0.07050349877798	1.311351310045

D.4.4 Parthasarathy results from four agents

Original		EpisodeTime	AvgTime/Game	Reward Total	Running/mean	Actual rewards		EpisodeTime	AvgTime/Game	Reward Total	Running/mean	Actual rewards
Set1	Mean Average	19.5312953333333	0.902053333333333	-20.4933333333333	-20.8393910533333	0.646666666666667	Total	48.8282383333333	2.25513333333333	-3074	-3125.908658	97
Set2	Mean Average	19.4562613333333	0.899819333333333	-20.5333333333333	-20.8456972733333	0.606666666666667	Total	48.6406533333333	2.24954833333333	-3080	-3126.854591	91
Set3	Mean Average	20.065128	0.930238666666667	-20.5866666666667	-20.4971494933333	0.553333333333333	Total	50.16282	2.32559666666667	-3088	-3074.572424	83
Set4	Mean Average	19.065194	0.880768	-20.5266666666667	-20.8426403066667	0.613333333333333	Total	47.662985	2.20192	-3079	-3126.396046	92
Set5	Mean Average	19.3230926666667	0.894937333333333	-20.58	-20.8905275066667	0.56	Total	48.3077316666667	2.23734333333333	-3087	-3133.579126	84
Set6	Mean Average	19.8971573333333	0.920658666666667	-20.5533333333333	-20.83092008	0.586666666666667	Total	49.7428933333333	2.30164666666667	-3083	-3124.638012	88
Set7	Mean Average	19.1972653333333	0.888148666666667	-20.5533333333333	-20.8459837066667	0.586666666666667	Total	47.9931633333333	2.22037166666667	-3083	-3126.897556	88
Set8	Mean Average	19.3304833333333	0.893502	-20.5533333333333	-20.3167765733333	0.586666666666667	Total	48.3262083333333	2.233755	-3083	-3047.516486	88
Set9	Mean Average	19.4828353333333	0.902701333333333	-20.58	-20.8771693333333	0.56	Total	48.7070883333333	2.25675333333333	-3087	-3131.5754	84
Set10	Mean Average	19.4379993333333	0.902026	-20.62	-20.89336694	0.52	Total	48.5949983333333	2.255065	-3093	-3134.005041	78
Mean Average	Original Deviance	0.286412682140965	0.013830851523557	0.033869028789015	0.186106282933424	0.033869026789015	Original Deviance	0.716031705352412	0.034577128808692	5.08035431835222	27.9159424400136	5.08035431835222
	Original Mean	19.4786712	0.901485333333333	-20.558	-20.7679622266667	0.582	Original Mean	48.696678	2.25371333333333	-3083.7	-3115.194334	87.3
	Original Mean Total	194.786712	9.01485333333333	-205.58	-207.679622266667	5.82	Original Total	486.96678	22.5371333333333	-30837	-31151.94334	873
	Original Deviation %	0.147039127669533	0.153422923392615	-0.016474865643066	-0.089612202151667	0.581942075412624	Original Deviation %	0.147039127669533	0.153422923392615	-0.016474865643066	-0.089612202151667	0.581942075412625

High Zero		EpisodeTime	AvgTime/Game	Reward Total	Running/mean	Actual rewards		EpisodeTime	AvgTime/Game	Reward Total	Running/mean	Actual rewards
Set1	Mean Average	19.225182	0.888765333333333	-20.54	-20.85698616	0.6	Total	48.062955	2.22191333333333	-3081	-3128.547924	90
Set2	Mean Average	18.9605	0.879913333333333	-20.6266666666667	-20.8728588733333	0.513333333333333	Total	47.40125	2.19978333333333	-3094	-3130.928831	77
Set3	Mean Average	19.4206093333333	0.894898	-20.4733333333333	-20.83586478	0.666666666666667	Total	48.5515233333333	2.237245	-3071	-3125.379717	100
Set4	Mean Average	20.0885846666667	0.929514666666667	-20.5466666666667	-20.80924262	0.593333333333333	Total	50.2214616666667	2.32378666666667	-3082	-3121.386393	89
Set5	Mean Average	19.2526726666667	0.894558	-20.64	-20.88957016	0.5	Total	48.1316816666667	2.236395	-3096	-3133.435524	75
Set6	Mean Average	19.5164766666667	0.902417333333333	-20.5333333333333	-20.8248495733333	0.606666666666667	Total	48.7911916666667	2.25604333333333	-3080	-3123.727436	91
Set7	Mean Average	19.1141653333333	0.884929333333333	-20.58	-20.8542104266667	0.56	Total	47.7854133333333	2.21232333333333	-3087	-3128.131564	84
Set8	Mean Average	18.929678	0.880855333333333	-20.6666666666667	-20.90176924	0.473333333333333	Total	47.324195	2.20213833333333	-3100	-3135.265386	71
Set9	Mean Average	19.5915786666667	0.906672	-20.5666666666667	-19.83559934	0.573333333333333	Total	48.9789466666667	2.26668	-3085	-2975.339901	86
Set10	Mean Average	19.037956	0.880706	-20.5533333333333	-20.8498981866667	0.586666666666667	Total	47.59489	2.201765	-3083	-3127.484728	88
Mean Average	Zero Deviance	0.336131300359481	9.2127810125691	0.054686988092842	0.306987270011399	0.054686988092842	Zero Deviance	0.840328250898701	0.036660802858397	8.20304821392633	46.0480905017098	8.20304821392633
	Zero Mean	19.3137403333333	0.894322933333334	-20.5726666666667	-20.753084936	0.567333333333333	Zero Mean	48.2843508333333	2.23580733333333	-3085.9	-3112.9627404	85.1
	Zero Mean Total	193.137403333333	8.94322933333334	-205.726666666667	-207.53084936	5.67333333333333	Zero Total	482.843508333333	22.3580733333333	-30859	-31129.627404	851
	Zero Deviation %	0.174037392321857	103.014030717418	-0.026582352681313	-0.147923680242292	0.963930459920838	Zero Deviation %	0.174037392321857	0.163971207678882	-0.026582352681313	-0.147923680242292	0.963930459920838

High Negative		EpisodeTime	AvgTime/Game	Reward Total	Running/mean	Actual rewards		EpisodeTime	AvgTime/Game	Reward Total	Running/mean	Actual rewards
Set1	Mean Average	19.1563866666667	0.884541333333333	-20.5133333333333	-20.8270397666667	0.626666666666667	Total	47.8909666666667	2.21135333333333	-3077	-3124.055965	94
Set2	Mean Average	19.2575053333333	0.886928	-20.4533333333333	-19.77210584	0.686666666666667	Total	48.1437633333333	2.21732	-3068	-2965.815876	103
Set3	Mean Average	18.9465533333333	0.877934	-20.58	-20.3235495266667	0.56	Total	47.3663833333333	2.194835	-3087	-3048.532429	84
Set4	Mean Average	19.27545	0.892982666666667	-20.5733333333333	-20.8370155333333	0.566666666666667	Total	48.188625	2.23245666666667	-3086	-3125.55233	85
Set5	Mean Average	19.029942	0.884440666666667	-20.6466666666667	-20.8934932933333	0.493333333333333	Total	47.574855	2.21110166666667	-3097	-3134.023994	74
Set6	Mean Average	18.599812	0.866394666666667	-20.6933333333333	-20.94156574	0.446666666666667	Total	46.49953	2.16598666666667	-3104	-3141.234861	67
Set7	Mean Average	19.5476126666667	0.90351	-20.5266666666667	-19.2856375866667	0.613333333333333	Total	48.8690316666667	2.258775	-3079	-2892.845638	92
Set8	Mean Average	19.0583306666667	0.885516666666667	-20.64	-20.8989708466667	0.5	Total	47.6458266666667	2.21379166666667	-3096	-3134.845627	75
Set9	Mean Average	19.2528966666667	0.891455333333334	-20.5733333333333	-20.3653146333333	0.566666666666667	Total	48.1322416666667	2.22863833333333	-3086	-3054.797195	85
Set10	Mean Average	18.6718433333333	0.866218	-20.6133333333333	-20.3380018466667	0.526666666666667	Total	46.6796083333333	2.165545	-3092	-3050.700277	79
Mean Average	Negative Deviance	0.272406505655807	0.010914275556149	0.067382820098631	0.527521878494979	0.067382820098631	Negative Deviance	0.681016264145176	0.027285888890373	10.1074230147946	79.128281474247	10.1074230147946
	Negative Mean	19.0796332666667	0.883992133333333	-20.5813333333333	-20.4482694613333	0.558666666666667	Negative Mean	47.6990831666667	2.20998033333333	-3087.2	-3067.2404192	83.8
	Negative Mean Total	190.796332666667	8.83992133333333	-205.813333333333	-204.482694613333	5.58666666666667	Negative Total	476.990831666667	22.0998033333333	-30872	-30672.404192	838
	Negative Deviation %	0.142773449494955	0.123465754327405	-0.032739773953079	-0.257978738735079	1.20613639794685	Negative Deviation %	0.142773449494955	0.123465754327405	-0.032739773953079	-0.257978738735079	1.20613639794685

High Positive		EpisodeTime	AvgTime/Game	Reward Total	Running/mean	Actual rewards		EpisodeTime	AvgTime/Game	Reward Total	Running/mean	Actual rewards
Set1	Mean Average	18.643944	0.865459333333333	-20.6333333333333	-20.38094408	0.506666666666667	Total	46.60986	2.16364833333333	-3095	-3057.141612	76
Set2	Mean Average	19.4751793333333	0.900855333333334	-20.54	-20.85203022	0.6	Total	48.6879483333333	2.25213833333333	-3081	-3127.804533	90
Set3	Mean Average	20.0381053333333	0.928364	-20.58	-20.8828293	0.56	Total	50.0952633333333	2.32091	-3087	-3132.424395	84
Set4	Mean Average	20.1501006666667	0.931734	-20.5466666666667	-20.86041954	0.593333333333333	Total	50.3752516666667	2.329335	-3082	-3129.062931	89
Set5	Mean Average	19.5089906666667	0.905869333333333	-20.62	-20.8836837	0.52	Total	48.7724766666667	2.26467333333333	-3093	-3132.552555	78
Set6	Mean Average	19.3943026666667	0.897466	-20.56	-20.8499217066667	0.58	Total	48.4857566666667	2.243665	-3084	-3127.488256	87
Set7	Mean Average	18.9885313333333	0.878336	-20.5533333333333	-20.85275428	0.586666666666667	Total	47.4713283333333	2.19584	-3083	-3127.913142	88
Set8	Mean Average	19.196722	0.888488666666667	-20.56	-20.3526209133333	0.58	Total	47.991805	2.22122166666667	-3084	-3052.893137	87
Set9	Mean Average	18.6809406666667	0.862978	-20.5533333333333	-20.36852684	0.586666666666667	Total	46.7023516666667	2.157445	-3083	-3055.279026	88
Set10	Mean Average	18.8830413333333	0.873122666666667	-20.5533333333333	-20.3137002	0.586666666666667	Total	47.2076033333333	2.18280666666667	-3083	-3047.05503	88
Mean Average	Positive Deviance	0.494544731818853	9.20801625630792	0.030147784145587	0.250437409927333	0.030147784145587	Positive Deviance	1.23636182954713	0.057488009339146	4.52216762183801	37.5656114890999	4.52216762183801
	Positive Mean	19.2959858	0.893267333333333	-20.57	-20.659743078	0.57	Positive Mean	48.2399645	2.23316833333333	-3085.5	-3098.9614617	85.5
	Positive Mean Total	192.959858	8.93267333333333	-205.7	-206.59743078	5.7	Positive Total	482.399645	22.3316833333333	-30855	-30989.614617	855
	Positive Deviation %	0.25629410020547	103.082424630341	-0.014656190639566	-0.121220002098679	0.528908493782222	Positive Deviation %	0.25629410020547	0.257428015976463	-0.014656190639566	-0.121220002098679	0.528908493782223

D.4.5 Parthasarathy results from original and negative RVMs across three sets (Last minute collection).

Original		Episode time	Average/game	Reward total	Running mean rewards	Actual Rewards		Episode time	Average/game	Reward total	Running mean rewards	Actual Rewards
Set1	Mean Average	20.92400067	0.968684667	-20.5733333	-20.87808176	0.566666666667	Total	3138.6001	145.3027	-3086	-3131.712264	85
Set2	Mean Average	19.31993333	0.89147	-20.5	-20.30160324	0.64	Total	2897.99	133.7205	-3075	-3045.240486	96
Set3	Mean Average	19.19409867	0.888174667	-20.5533333	-20.3740344066667	0.586666666667	Total	2879.1148	133.2262	-3083	-3056.105161	88
Original	Original Total	59.43803267	2.748329333	-61.6266667	-61.5537194066667	1.793333333333	Original Total	8915.7049	412.2494	-9244	-9233.057911	269
Original	Original Mean	19.81267756	0.916109778	-20.5422222	-20.5179064688889	0.597777777778	Original Mean	2971.901633	137.4164666667	-3081.33333	-3077.68597033333	89.6666666666667
Original	Original Deviance	0.787501485	0.037200394	0.030951974	0.256393250979042	0.030951973949	Original Deviance	118.1252228	5.580059155202	4.642796092	38.4589876468564	4.64279609239471
Original	Original Deviance %	1.324911761	1.353563924	-0.05022497	-0.416535756816139	1.725946502749	Original Deviance %	1.324911761	1.353563923975	-0.05022497	-0.416535756816141	1.72594650274896

Negative		Episode time	Average/game	Reward total	Running mean rewards	Actual Rewards		Episode time	Average/game	Reward total	Running mean rewards	Actual Rewards
Set1	Mean Average	19.260614	0.894201333	-20.6266667	-20.8765732333333	0.513333333333	Total	2889.0921	134.1302	-3094	-3131.485985	77
Set2	Mean Average	19.437788	0.899717333	-20.5666667	-20.85617592	0.573333333333	Total	2915.6682	134.9576	-3085	-3128.426388	86
Set3	Mean Average	19.41669	0.893789333	-20.4533333	-20.81163268	0.686666666667	Total	2912.5035	134.0684	-3068	-3121.744902	103
Negative	Negative Total	58.115092	2.687708	-61.6466667	-62.5443818333333	1.773333333333	Negative Total	8717.2638	403.1562	-9247	-9381.657275	266
Negative	Negative Mean	19.37169733	0.895902667	-20.5488889	-20.8481272777778	0.591111111111	Negative Mean	2905.7546	134.3854	-3082.33333	-3127.21909166667	88.6666666666667
Negative	Negative Deviance	0.079018612	0.002702616	0.071870941	0.027115854501576	0.071870940572	Negative Deviance	11.85279183	0.405392353159	10.78064109	4.06737817523642	10.7806410858642
Negative	Negative Deviance %	0.135969177	0.100554662	-0.11658528	-0.043354580710117	4.052872588671	Negative Deviance %	0.135969177	0.100554661731	-0.11658528	-0.043354580710117	4.05287258867073

Frequencies

	0	1	2	3	4	5	6
Original	265	128	42	16	1	1	0
Negative	276	118	35	18	6	0	0

APPENDIX E – TECHNICAL REPORT

Table of Contents

List of Figures

List of Tables

ABSTRACT

To propose an improved reward accumulation whilst minimising negative impact in RL solutions, this study empirically explores effects of immediate long-term reward combinations (ILRC) by introducing immediate reward value modifications (RVM) to current known solutions.

Using the cross-entropy method (CEM) and two different policy gradient (PG) implementations, immediate rewards are implemented within each solution with differing values, rewards where rewards are not issued, negative rewards and positive rewards improving the original solutions performance in reward accumulation to satisfy the reward hypothesis. Collecting 10 discreet simulations of 150 episodes for each implementation with a quantitative-qualitative approach, this study yields an average 33% increase in reward accumulation across all ILRCs sufficient to minimise negative side effects by a difference of 5% improvement compared to original solutions.

With an average 3% in standard deviation across all three original implementations and an average 7% standard deviation across all three solutions with ILRCs, leaves an average difference of 2% reduction in standard deviation for each ILRC type. This reduction demonstrates better consistency in achieving performance improvements and a reduction of negative effects. The study exposed high value inhibitors (penalising for incorrect actions) to present most promising results outperforming ILRCs for neutral and positive rewards by an average 12.5%.

These results reflect on the importance of exploring the behavioural traits of ‘intelligent’ machines and the effects that are exhibited to facilitate improvements to their implementations in preparation for introduction into engineered autonomous systems and into society. Improper use and lack of information about autonomy, can have profound effects when operating autonomously coupled with machines in a real-world scenario. Thus, improving the operational capability and safety of autonomous systems is essential.

1 INTRODUCTION

An empirical approach has been taken to explore combining reward values in reinforcement learning (RL) solutions to understand the effects exhibited in artificial Intelligent (AI) behaviours, while solving a given problem like playing a game of Pong against an opponent or balancing a Pole on a Cart for longest time possible. Focussing on immediate and long-term reward combination (ILRC), several quantitative-qualitative methods have been used to generate quantifiable data to satisfy several objectives and hypotheses following this statement;

"We don't currently have any rules for how robots should behave if and when they start operating autonomously" (Wakefield 2015).

The approach to combine immediate rewards (preferred by animals) and long-term (probability of behaviour now, effecting future reward), stems from behavioural traits observed in human society today, reward for good behaviour, punishing poor behaviour and other combinations such as the quantity of reward and rewarding for noncontribution. Three hypotheses have been derived from this societal introspect:

i. Logical combinations of Reward Value Modifications (RVM) may positively affect the agent' actions on the environment, if immediate rewards were gained for actions ahead of final reward value.

ii. Immediate rewards may not prove to possess efficiency, scalability, safety, ethical or 'general-purpose' requirements to satisfy introduction into society.

iii. Combinations of high negative or high positive reward values may not be suited to specific algorithms nor prove effective for all tasks without solutions.

Three secondary sources of known RL solutions have been used as testbeds for this study, a cross entropy method (CEM) by Frans (2016a) and two policy gradient (PG) implementations by Karpathy (2016a) and Parthasarathy (2016) whose implementations make use of neural-nets (NN). Across three RL solutions, it is shown high negative values for incorrect actions, have significant impact on performance enhancement. However, the full effects of this enhancement on larger scales or when applied to mechanically engineered solutions have not been explored due to time and resource limitation.

1.1 PROBLEM DEFINITION AND REQUIREMENTS

Machine Learning (ML) should meet several operational, ethical and legislative requirements (UK Government 2015, 2016; European Parliament 2016; IEEE 2016) to be deemed worthy for evaluating their success for application in engineered autonomous solutions. Without meeting them, the safety of society and its inhabitants are put at risk. Thus, the importance of studying the implications and effects of modifications to known and future potential solutions, is fundamental.

The problems found while attempting to engineer ML solutions primarily stem from learning in dynamic environments and task variance where ethical, operational and legislative requirements must be satisfied to complete complex tasks and ensure the safety of others. Some concrete AI safety problems are defined by the OpenAI organisation (Brochman & Christiano 2016);

Problem	Problem example
Safe Exploration	Learning about the environment without causing issues
Robustness in distribution shift	Adaptability for changing data and safe failover
Avoiding negative side effects	Modifying the reward function without negative impact
Avoiding reward hacking	Creating more mess to satisfy cleaning rewards
Scalable oversight	With minimal interaction from a human, can a robot still perform specified task to satisfaction

Table 1 Concrete AI safety problems addressed by OpenAI (Adapted from Brochman & Christiano 2016).

RL is based on the reward hypothesis “All goals can be described by the maximisation of expected cumulative reward” (Silver 2015a); developing a robust, scalable and predictable autonomous system which satisfies this hypothesis and basic safety requirements is essential. Avoiding negative effects is the primary focus of this study.

1.2 OBJECTIVES

Using the two PG implementations enables a direct comparison of one to the other, initially on time taken but also for initial variance in performance. CEM adds diversity in the group and though it would be preferred to compare all known solutions and algorithms, the allotted timeframe did not permit for this. The purpose of this report is to examine results from studies into the effects of ILRC reward values on behaviour in RL implementations (agents).

The objectives are simple;
Source, examine and implement several known RL algorithms which may be applicable in extrapolating sufficient results to determine the effects of Reward Value Modification (RVM) introduced to effect ‘intended’ behaviour of the original source code.

Collect and analyse sufficient data to represent the effects and trends RVM implementations have had on Agent’s performance, in comparison to the original implementation and to each other.

1.3 INTRODUCTION TO CEM AND PG

CEM is a hill-climbing algorithm, starting with a random variable, the policy increments by a constant until an optimum solution is reached, the incremental values should preferably start with a low value, 0.1 for example (Frans 2016b).

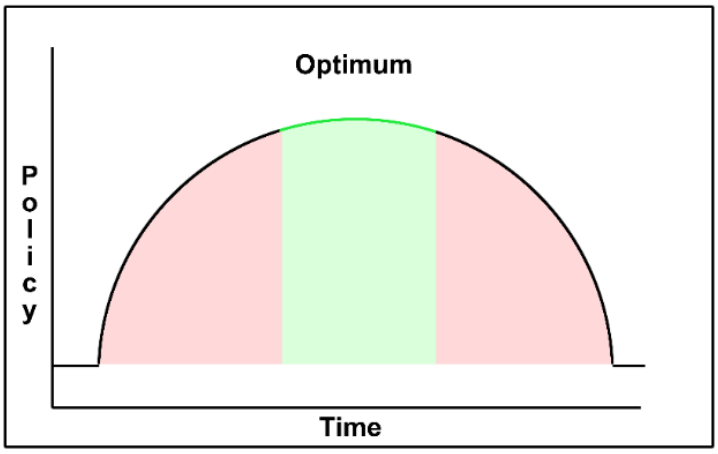


Figure 1 CEM algorithm increments policy value until optimum is reached, too high and optimum is overshot, too low and policy does not start incrementing (Amended from Frans 2016b).

PG makes use of action probability based on discreet samples of actions performed. If an action of moving the paddle UP produced better results than moving DOWN in Atari Pong for example in a given state of the game, the gradient updates the NN weights and probability distribution over actions in any given state and continues to do so until sufficient training leads to optimum probability of action performance to maximise expected long-term reward (Silver 2015a, 2015b; Karpathy 2016b).

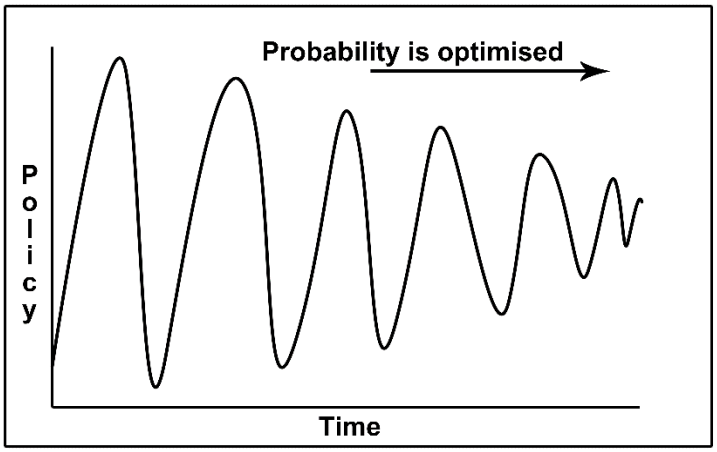


Figure 2 PG algorithm improves the probability of actions performed, gaining higher long-term rewards. Can take considerably longer than CEM and requires more training (Adapted from Karpathy 2016b).

Combining these features with immediate rewards on actions performed, may present better performance and could improve the convergence time for long-term optimal performance of a solution over its predecessor.

2 FINDINGS

As the discussion and exploration of results continues, numeric values have been rounded to 1 decimal place for simplifying representation though, graphs use higher decimal place values for accuracy and Table 2 shows the RVMs applied to each ILRC. Original solutions have no immediate rewards allocated to them. The positive ILRCs have high valued immediate rewards for positive actions (actions which gain long term rewards or win a game). The negative ILRCs have high valued inhibitors (immediate rewards taken away from the accumulated rewards in an episode) and the neutral ILRCs are given rewards for gaining rewards for actions which do not generally merit a reward. The single values 1 and -1, represent a small reward for that action.

Developer / ILRC RVM	Original	Positive	Negative	Neutral
Frans	None	0 == 0 Reward == 200 -Reward == -1	0 == 0 Reward == 1 -Reward == -200	0 == 200 Reward == 1 -Reward == -1
Karpathy	None	0 == 0 Reward == 200 -Reward == -1	0 == 0 Reward == 1 -Reward == -200	0 == 200 Reward == 1 -Reward == -1
Parthasarathy	None	0 == 0 Reward == 200 -Reward == -1	0 == 0 Reward == 1 -Reward == -200	0 == 200 Reward == 1 -Reward == -1

Table 2 Shows the values assigned to the immediate rewards for each Agent type per developer solution (Author 2017).

2.1 CEM RESULTS

Since this study used CEM to test the RVM functionality and was used first to gather results, it seems apt as a starting point. The CEM algorithm was implemented using CartPole environment (see Figure 3), where the goal is simply to develop a solution which balances the Pole on a Cart for longest time possible on a frictionless track. Episodes reset if the maximum time is reached (set to 200 timesteps equal to 200 rewards), the Pole tips more than 15 degrees from vertical or the Cart moves more than 2.4 units (OpenAI 2016). The problem is considered solved if the Pole can be balanced for 135 timesteps or more.

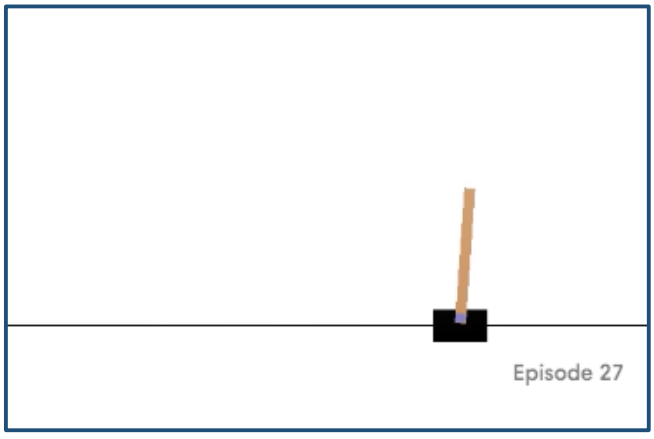


Figure 3 Cart and Pole environment. A Cart must balance a Pole for longest time possible on a frictionless track, episode ends if the crat moves more than 2.4 units or if the Pole tips more than 15 degrees (OpenAI 2016).

2.1.1 Observations

As a discreet sample, from each RVM type (including the original with no RVMs) the first 150 episodes from set 1 of 10 are taken for sampling the observation data (see Figures 4 to 7). The observation data shows us the directional movement of the Cart and Pole in the environment.

The original implementation is making smaller adjustments to compensate for movement in comparison to the other three implementations. What this shows is the three ILRC implementations are making more ‘informed decisions’ over which actions to take, resulting in longer durations for each episode hence the distribution of waveforms. Furthermore, when comparing negative to original, the negative observations reflect improved adaptation in response to the Pole movement over time.

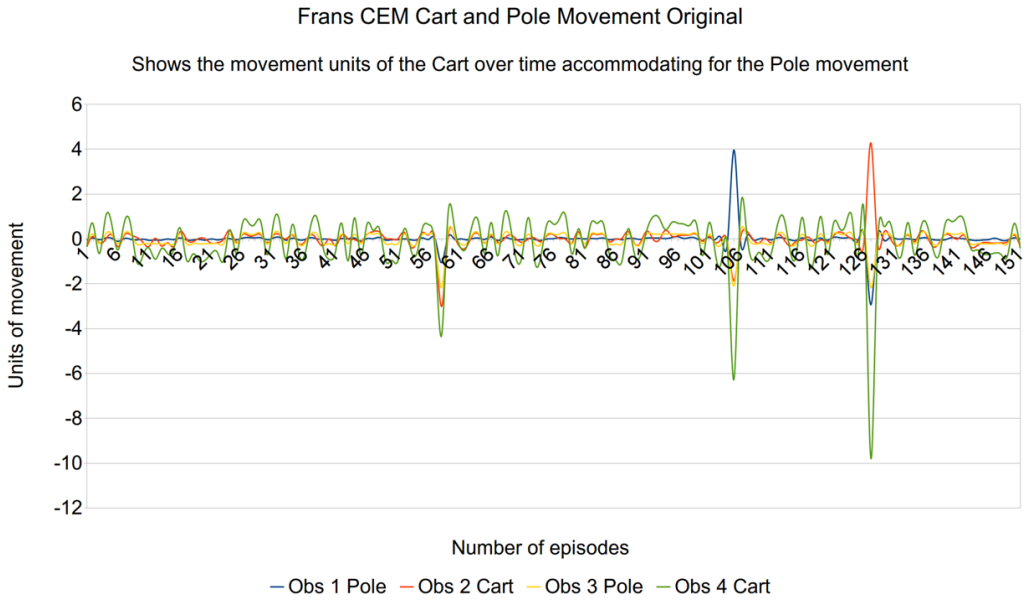


Figure 4 Sampled environment observation data from Original CartPole CEM (Author 2017).

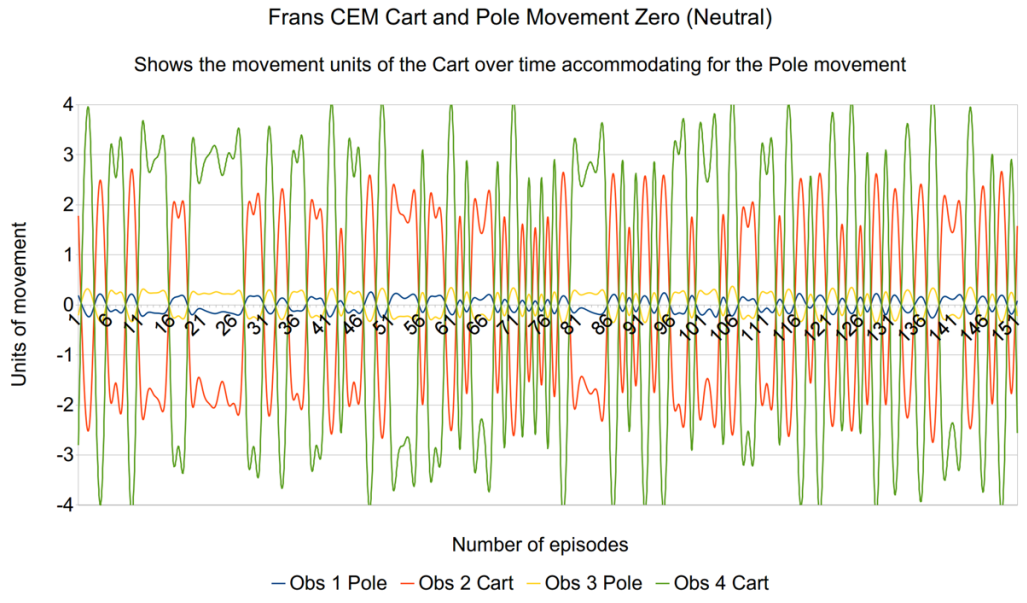


Figure 5 Sampled environment observation data from Neutral ILRC CartPole CEM (Author 2017).

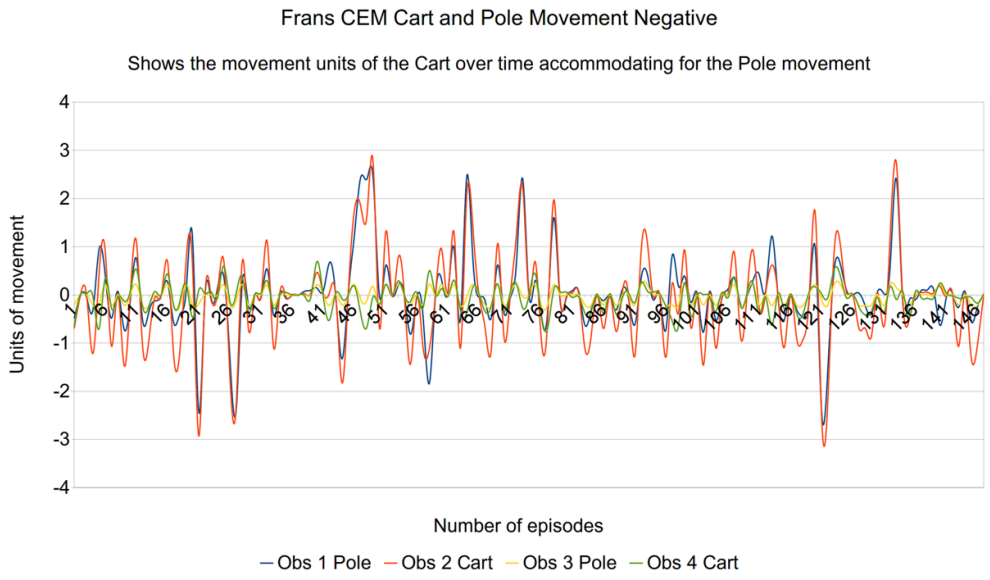


Figure 6 Sampled environment observation data from Negative ILRC CartPole CEM (Author 2017).

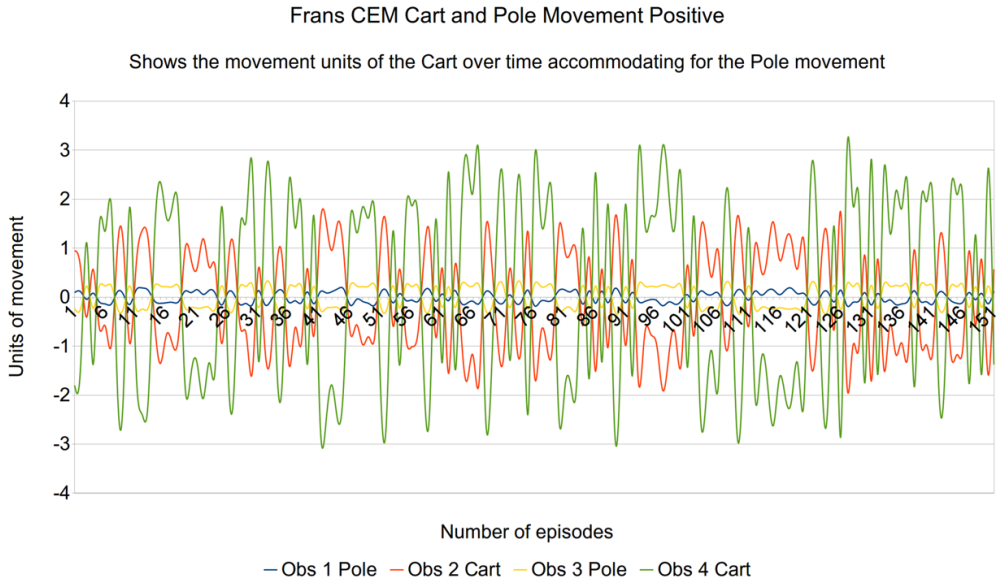


Figure 7 Sampled environment observation data from Positive ILRC CartPole CEM (Author 2017).

2.1.2 Total rewards

The observations in Figures 4 to 7 are reflected when studying the total rewards gained over all episodes where a mean average of 16.7% difference in reward gain is observed, compared to the original implementation (see Tables 3 and 4). It is clear from the differences; large inhibiting values improve the performance for reward gain over time in CEM outperforming the other ILRCs.

Agent Type / ILRC	Rewards gained across 1500 episodes	% Difference from original
Original	78856.72	0
Zero	73000.00	-7.43
Negative	159418.00	102.16
Positive	43747.00	-44.52
Mean difference between RVMs and original		16.74

Table 3 Total rewards accrued in 1,500 episodes across four agents and the difference of the original in each RVM.

Agent Type / ILRC	Average Rewards gained / episode across 1500 episodes	% Difference from original
Original	52.57	0
Zero	48.66	-7.44
Negative	106.27	102.15
Positive	29.16	-44.53
Mean difference between RVMs and original		16.73

Table 4 Compares the average rewards per episode and percentile differences of the original implementation for RVMs.

2.1.3 Average time

An increase in time taken to complete an episode is also exhibited when reviewing the negative ILRC, to be expected if the Agent is improving its performance since the episode must last longer to improve rewards.

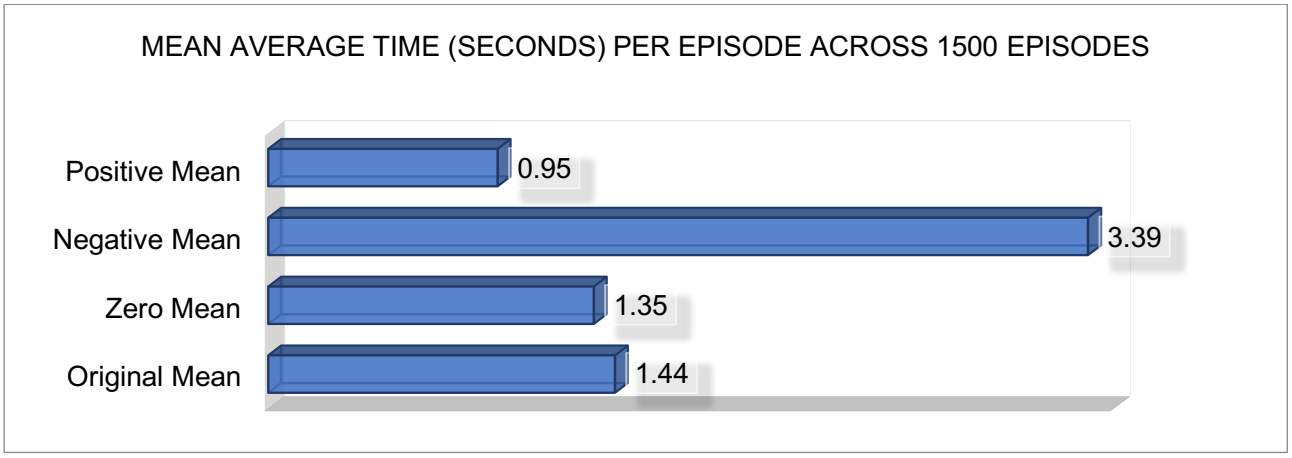
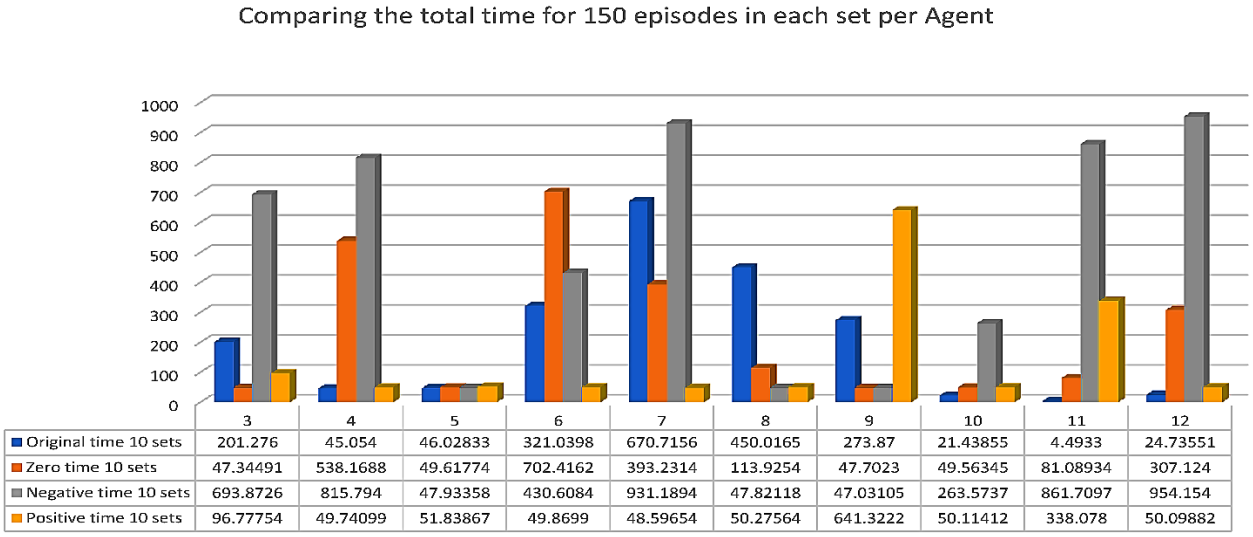


Figure 8 Mean average time (Seconds) per episode across 1,500 episodes for all implementations.



	3	4	5	6	7	8	9	10	11	12
Original time 10 sets	201.276	45.054	46.02833	321.0398	670.7156	450.0165	273.87	21.43855	4.4933	24.73551
Zero time 10 sets	47.34491	538.1688	49.61774	702.4162	393.2314	113.9254	47.7023	49.56345	81.08934	307.124
Negative time 10 sets	693.8726	815.794	47.93358	430.6084	931.1894	47.82118	47.03105	263.5737	861.7097	954.154
Positive time 10 sets	96.77754	49.74099	51.83867	49.8699	48.59654	50.27564	641.3222	50.11412	338.078	50.09882

Figure 9 Compares the total time (Seconds) for 150 episodes per Agent set.

2.1.4 Standard Deviation

Standard Deviation (SD) permits measurement of consistency in specified results from a population of samples. With lower values demonstrating more consistency from a population

sample. High negative reward values have shown to improve the SD when compared to the original implementation. Although the SD is low on average across the board, values for zero reward and high valued positive reward only show a marginally higher deviation than the original reflecting a slight reduction in performance. Representing the deviation across 1,500 episodes, the percentile unit in Figure 10 and 11 reflects SD percentage obtained by dividing SD by mean average from all episodes.

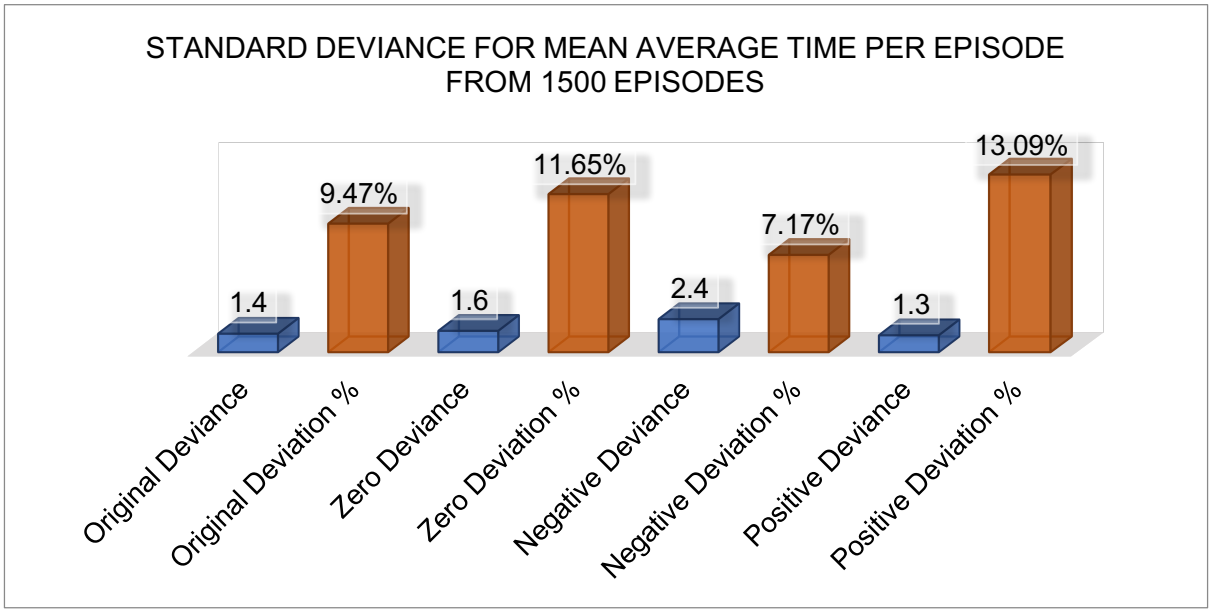


Figure 10 Mean Time Standard Deviance across 1,500 episodes. Percentages derived from average time per episode.

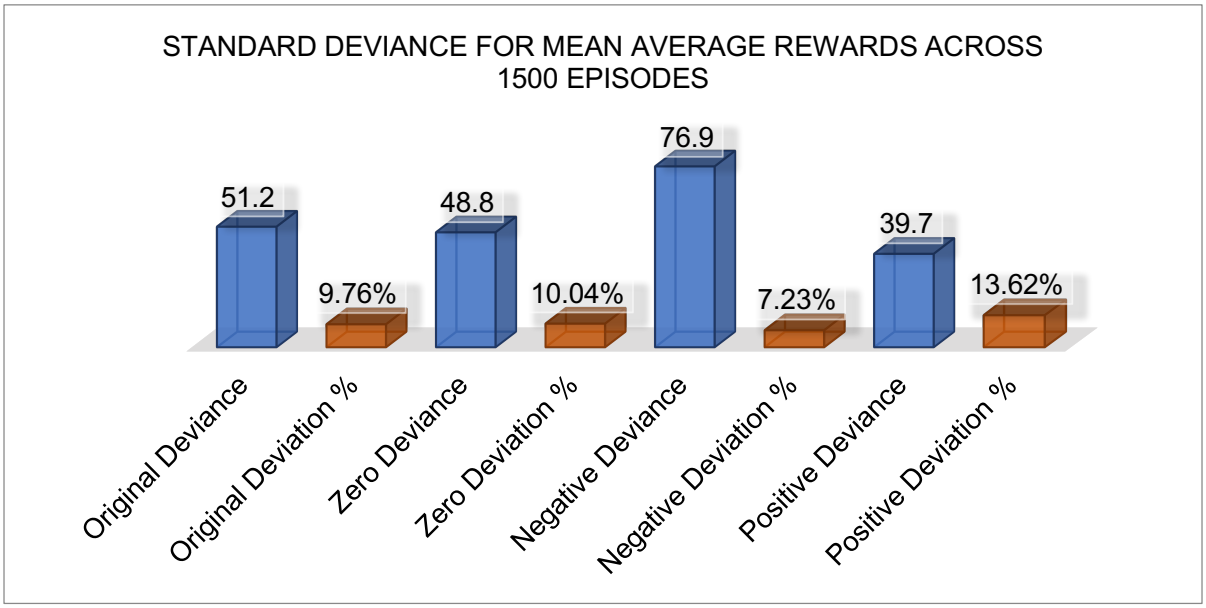


Figure 11 Mean reward Standard Deviance across 1,500 episodes. Percentages derived from average rewards episode.

2.2 CEM Summary

Results from 1,500 episodes for each Agent, have shown immediate rewards applied to a CEM solution, outperform the original solution with an average of 16.7% difference.

Of these RVMs, high valued negative rewards for incorrect actions produce -25.9% difference in standard deviation, representing 25.9% increase in consistency for reward gain. Although the deviations are low, they should preferably be 0% which reflects 100% robustness in solving the problem from the start of the trial. However, upon reflection humans are never 100% consistent.

Furthermore, when comparing the episode times for original CEM solution, the difference of 175.3% for negative ILRC episode time appears unacceptable. However, this reflects longer interaction with the environment and given the goal of CartPole environment, the increase in time reflects higher gain in rewards, evident from an increased difference of 102% exhibited by the negative ILRC when compared to the original solution.

2.3 KARPATHY PG RESULTS

Implementations by Karpathy and Parthasarathy, use the Atari Pong game environment. The goal is to maximise score by hitting a ball past the opponent with a paddle using actions UP or DOWN. Episodes are reset after 21 games are won by either opponent. Both implementations from each developer are slightly different in approach particularly where they update the NN weights.

2.3.1 Total rewards

All three RVMs outperform the original solution though, marginally; when compared to CEM, negative reward values are still presenting marked improvements above the zero and positive rewards.

Assessing the average rewards gained per episode across 1,500 episodes, shows a mean average of 7.5% difference in reward gain per episode across all three ILRCs compared to the original implementation and despite smaller margins compared to CEM.

Agent Type / ILRC	Rewards gained across 1500 episodes	% Difference from original
Original	836	0
Zero	876	4.78
Negative	937	12.08
Positive	894	6.94
Mean difference between RVMs and original		7.93

Table 5 Compares the total rewards gained in 1,500 episodes for Karpathy PG across four Agents.

Agent Type / ILRC	Average Rewards gained / episode across 1500 episodes	% Difference from original
Original	0.56	0
Zero	0.58	3.44
Negative	0.62	10.71
Positive	0.6	7.14
Mean difference between RVMs and original		7.1

Table 6 Compares the average rewards gained per episode for four Agents across 1,500 episodes.

2.3.2 Average time

Despite negative RVM having an 11.4% average increase in reward gain over the original implementation, a 0.56% increase in time per episode is observed. The time increase from PG negative ILRC is minimal when compared to the 175.3% increase for the same ILRC in CEM. In

addition, an average reward ratio of 10:1 in favour of CEM, confirming Karpathy' conclusion (2016b) of CEM performance over PG.

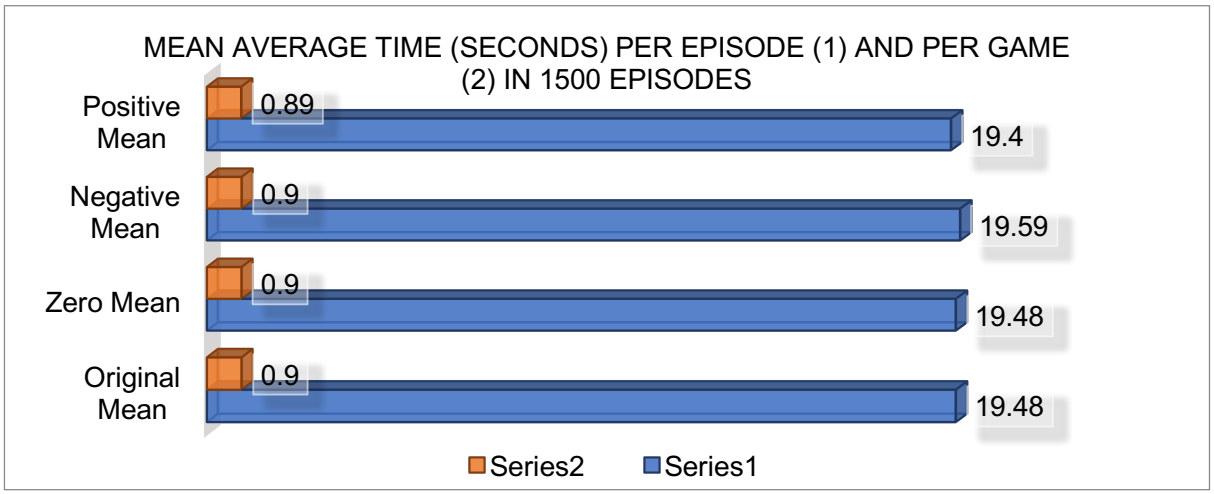
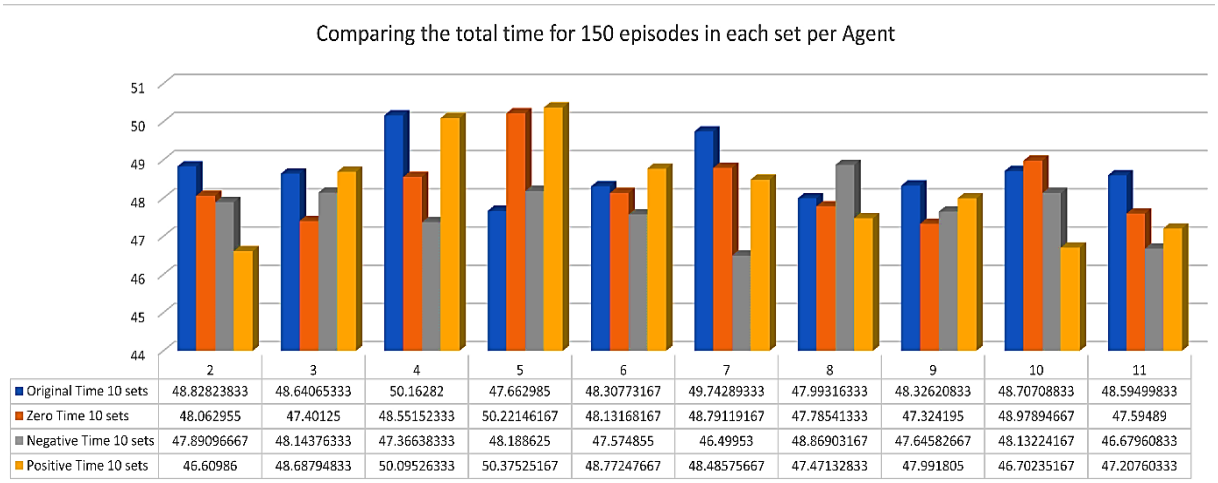


Figure 12 Mean average time (Seconds) per episode and per game across 1,500 episodes for all implementations.



	2	3	4	5	6	7	8	9	10	11
Original Time 10 sets	48.82823833	48.64065333	50.16282	47.662985	48.30773167	49.74289333	47.99316333	48.32620833	48.70708833	48.59499833
Zero Time 10 sets	48.062955	47.40125	48.55152333	50.22146167	48.13168167	48.79119167	47.78541333	47.324195	48.97894667	47.59489
Negative Time 10 sets	47.89096667	48.14376333	47.36638333	48.188625	47.574855	46.49953	48.86903167	47.64582667	48.13224167	46.67960833
Positive Time 10 sets	46.60986	48.68794833	50.09526333	50.37525167	48.77247667	48.48575667	47.47132833	47.991805	46.70235167	47.20760333

Figure 13 Compares the total time (Minutes) for 150 episodes per Agent set.

2.3.3 Standard Deviation

When comparing CEM and Karpathy' PG, similar trends in deviation are reflected when reviewing the correlation between high performing ILRCs and their SDs. These results show as the performance increases from the modification, as does the efficiency and consistency in accumulating rewards.

However, this trend does not seem to follow the same path for rewards across PG implementations since, the SD does not decrease over the same discreet populations for PG where all ILRCs outperformed the original the SD representing lower consistency and robustness, albeit marginally (see Figure 15). Notice negative ILRC had the highest reward accumulation (Tables 5 and 6) and the SD is higher than the original (Figure 15) reflecting marginally poorer performance compared to the original solution.

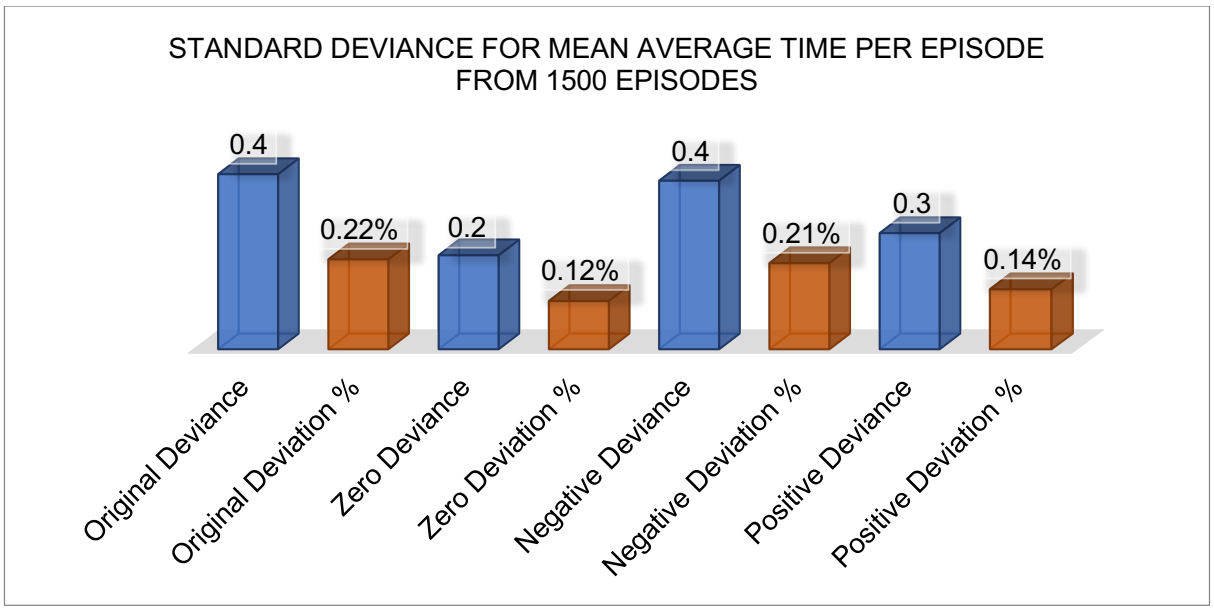


Figure 14 Mean Time per episode Standard Deviance across 1,500 episodes.

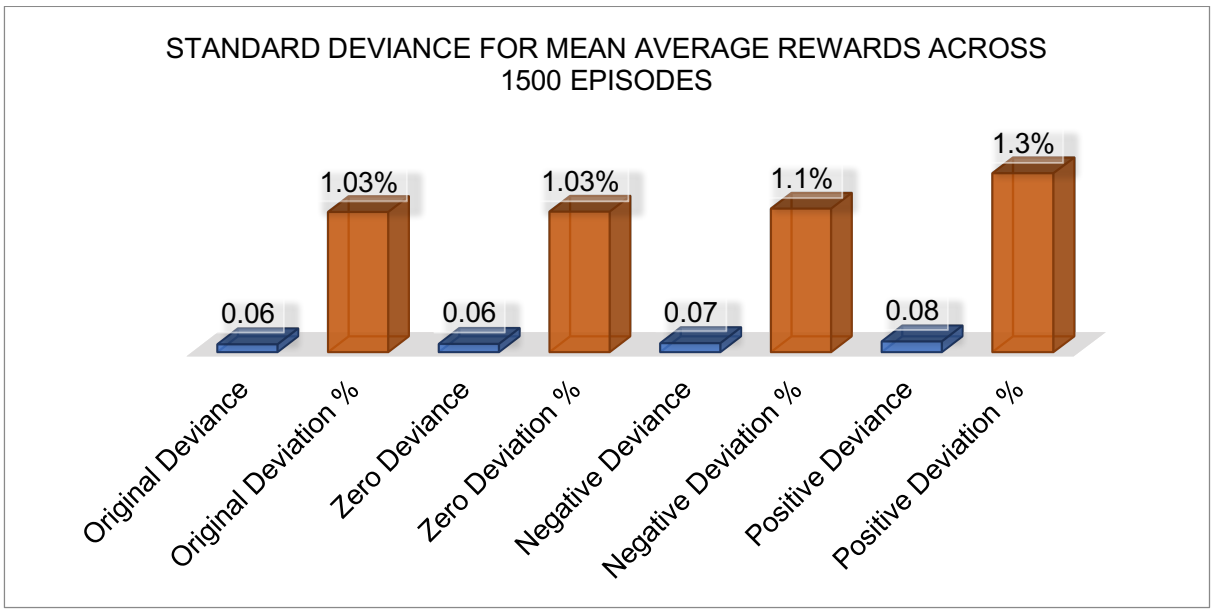


Figure 15 Mean reward Standard Deviance across 1,500 episodes.

2.4 SUMMARY

Results from 10 simulations of 150 episodes for each Agent, have shown immediate rewards applied to a PG solution, outperform the original solution across all RVMs and by an average difference of 7.5% (lower performance than CEM). However, while negative RVMs showed the longest interaction time and their reward SD was low, they did not produce better SD than the original solution. The mean average SD for the PG ILRCs is only 0.07% for reward accumulation which is 0.01% lower in robustness compared to their original predecessor. However, this is still a good SD considering the preferred is 0% deviation.

Additionally, the time taken for an episode of Pong to finish, does not necessarily reflect improved reward for example; if the time taken to compete was 20 minutes where each opponent scored a

point every 30 seconds, and in a second game, the time taken was 10 minutes where one opponent scored 20 reward and other did not, the average deciding factor is 30 seconds meaning, the variation in time is not heavily weighted on reward since, time and rewards are separate factors to be measured.

Furthermore, the ratio of negative ILRC reward gain for PG in comparison to CEM is 10:1 in favour of CEM. However, despite these differences current trends are demonstrating higher performance through immediate rewards when heavily penalised for incorrect actions. It should be noted however, the PG solutions used are untrained and better performance may be exhibited with trained or improved solutions.

2.5 PARTHASARATHY PG RESULTS

Parthasarathy solution did not yield similar trends observed in Karpathy and Frans implementations, in fact they present quite the opposite. A previous modification to an epsilon value (epsilon value was found to control the NN weight updates) was overlooked and the study had only enough time to gather another three sets after this was noticed.

2.5.1 Total rewards

The results from Parthasarathy' solution do not exhibit similar trends when compared to CEM and Karpathy' PG. Across all three ILRCs, not one has outperformed the original. These differences are marginal with an average difference of 3.2% decrease for reward accumulation. These differences may be explained by the epsilon value modified during exploration and overlooked when this study was performed.

Agent Type / ILRC	Rewards gained across 1500 episodes	% Difference from original
Original	873	0
Zero	851	-2.52
Negative	838	-4.01
Positive	855	-2.06
Mean difference between RVMs and original		-2.86

Table 7 Compares the total rewards gained in 1,500 episodes for Parthasarathy PG across four Agents.

Agent Type / ILRC	Rewards gained across 1500 episodes	% Difference from original
Original	0.58	0
Zero	0.56	-3.45
Negative	0.55	-5.17
Positive	0.57	-1.72
Mean difference between RVMs and original		-3.45

Table 8 Compares the average rewards gained per episode for four Agents across 1,500 episodes.

The same study was performed when resetting the epsilon to its original value (1e-5). However, only three simulations were able to be run for original and negative Agents which should be enough to see differences in the two different implementations and compare the other solutions (CEM and Karpathy' PG). The values in Tables 9 and 10 hold the actual values from three simulations and approximate values of 10 simulations. The approximate values are generated by

dividing the actual value by simulation quantity (3) and multiplied by 10. This represents similar values held in Tables 7 and 8 which were gathered from 10 simulations.

Agent Type / ILRC	Approximate rewards gained across 1500 episodes	Original value	% Difference from original
Original	896.66	269	0
Negative	886.66	266	-1.12

Table 9 Compares the total rewards gained in 1,500 episodes for Parthasarathy PG across two Agents without modified epsilon value.

Agent Type / ILRC	Approximate rewards gained across 1500 episodes	Original value	% Difference from original
Original	2	0.6	0
Negative	1.96	0.59	-2

Table 10 Compares the average rewards gained per episode for two Agents across 1,500 episodes without modified epsilon value.

Comparing the original ILRC with the first trial an average difference of 3% decrease for reward accumulation is measured and a 1.5% decrease in the second trial, the second trial has outperformed the modified epsilon with a difference of 1.5%. However, neither ILRCs have outperformed the original. Based on the trends exhibited by the first trial (Tables 7 and 8, epsilon modified), the trends exhibited by Karpathy PG and Frans CEM may still translate to Parthasarathy ILRCs given more time to run the simulations and examine ILRC results without the epsilon modification.

2.5.2 Average time

As to be expected when considering other trends in time across CEM and PG, a decrease in performance also manifests itself where the average time taken is less when fewer rewards are accumulated (see Figures 16 and 17). When comparing the time taken for negative ILRC in the first trial, there is 0.2% decrease in time taken per episode where Karpathy' solution was a 0.56% increase.

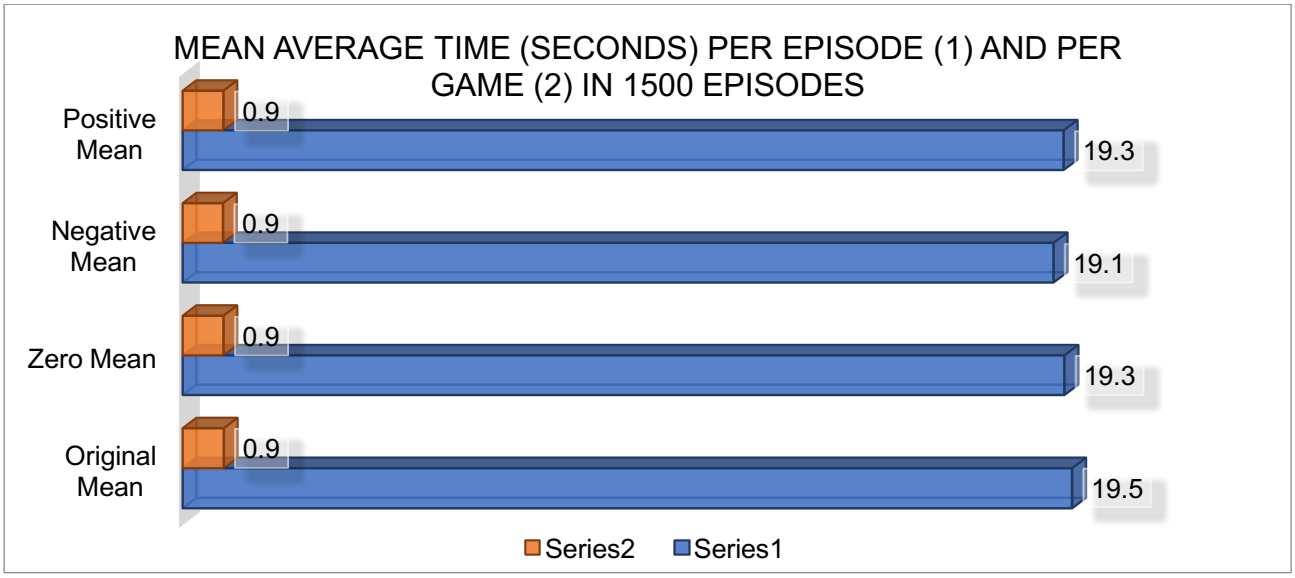
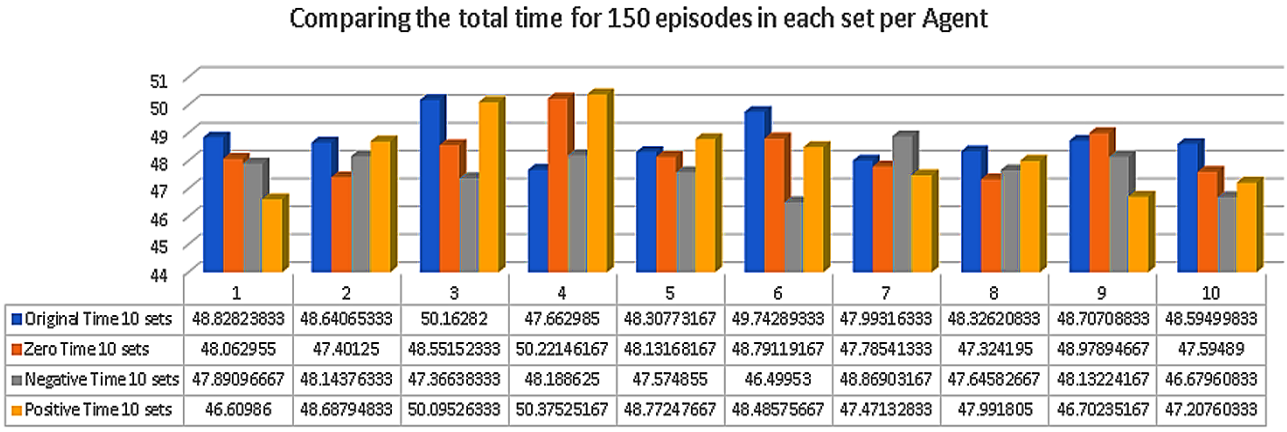


Figure 16 Mean average time (Seconds) per episode and per game across 1,500 episodes for all implementations.



	1	2	3	4	5	6	7	8	9	10
Original Time 10 sets	48.82823833	48.64065333	50.16282	47.662985	48.30773167	49.74289333	47.99316333	48.32620833	48.70708833	48.59499833
Zero Time 10 sets	48.062955	47.40125	48.55152333	50.22146167	48.13168167	48.79119167	47.78541333	47.324195	48.97894667	47.59489
Negative Time 10 sets	47.89096667	48.14376333	47.36638333	48.188625	47.574855	46.49953	48.86903167	47.64582667	48.13224167	46.67960833
Positive Time 10 sets	46.60986	48.68794833	50.09526333	50.37525167	48.77247667	48.48575667	47.47132833	47.991805	46.70235167	47.20760333

Figure 17 Compares the total time (Minutes) for 150 episodes per Agent set.

2.5.3 Standard Deviation

Although negative ILRC modification shows the worst performance over both trials, its difference in reward accumulation still exhibits marked improvements with an average difference of 2.9% for the first trial. Although, the time was slower compared to the original (see Figure 16).

Considering SD compares variance from the average norm, time duration is irrelevant when comparing the SD from a discreet population of samples evident when reviewing the reward accumulation SD (see Figure 19).

The SDs for both negative and positive ILRC modifications, exhibit mirrored shift. Where positive ILRC modifications performed better in rewards accumulation compared with negative. Positive ILRC time SD was higher than original and the opposite was true for negative ILRC. The problem faced here, is which of the two ILRCs would be best matched to outperform the original. Further simulations may be required to find an equilibrium of ILRC RVMs. Perhaps the solution may be another NN layer to assess the optimum value for immediate rewards in anticipation for long-term reward maximisation.

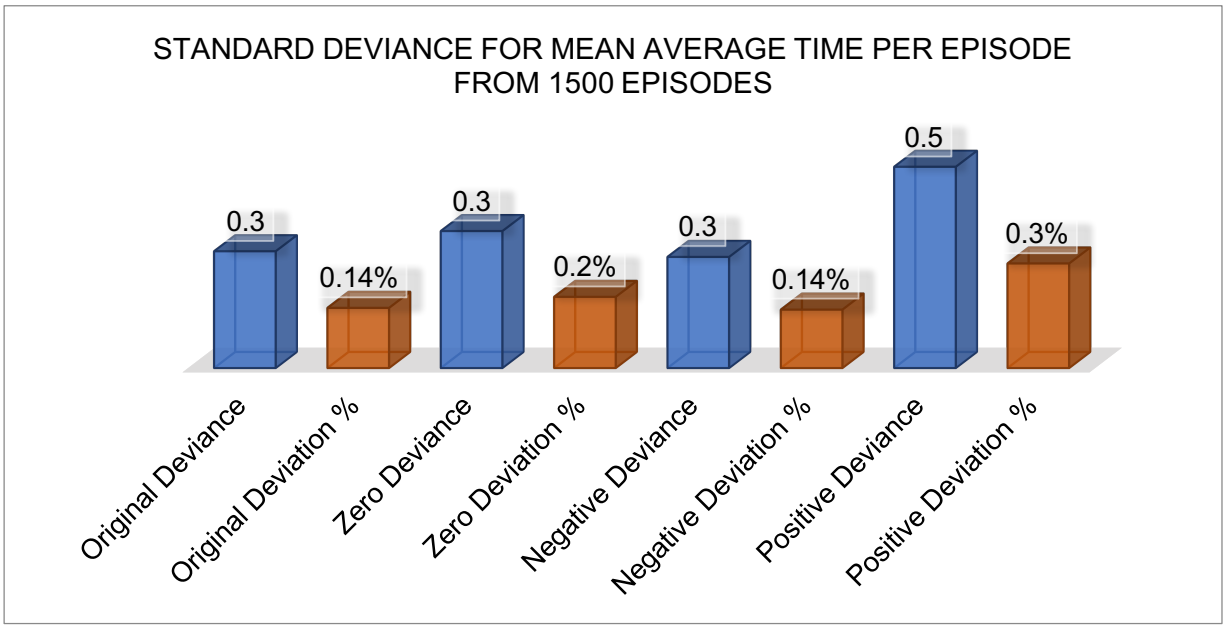


Figure 18 Mean Time Standard Deviance across 1,500 episodes.

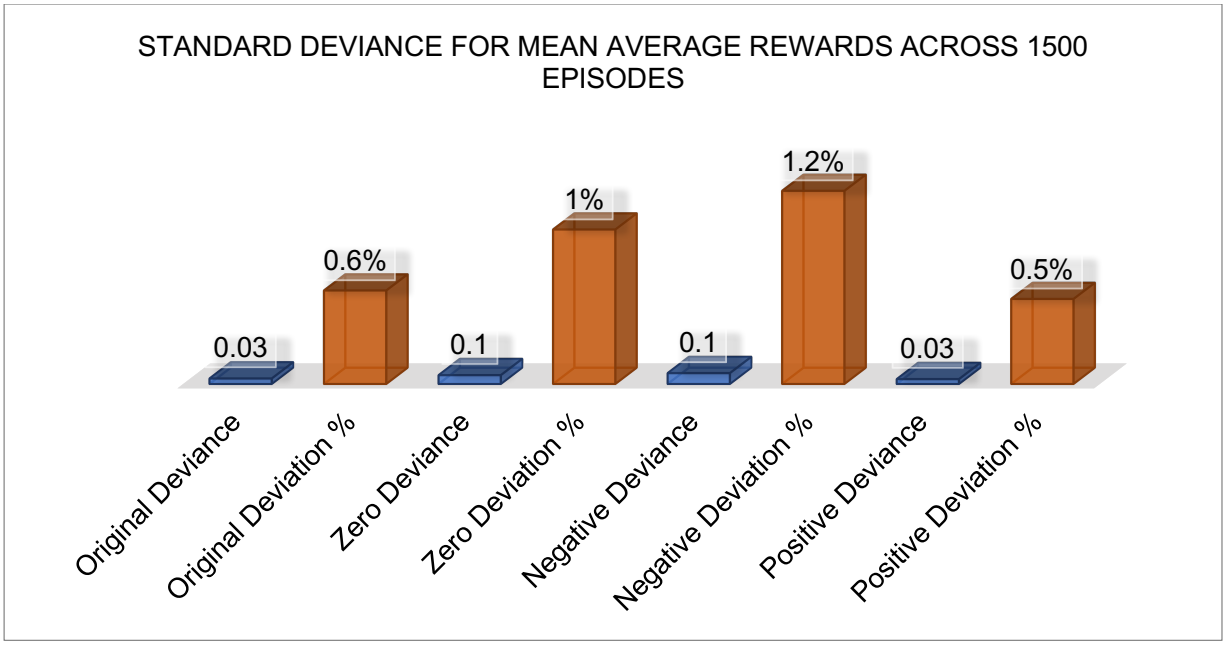


Figure 19 Mean reward Standard Deviance across 1,500 episodes.

2.6 SUMMARY

While both trials from Parthasarathy' solution have an average 1.5% difference in reward accumulation, the modified epsilon value has not improved the performance over the original value nor the approximate performance in the second trial (original epsilon value) compared to the original solution.

Another trial will be required to assess all ILRCs against the original solution since, the results from both trials are inconclusive especially when factoring in comparison benchmarks from trends exhibited in CEM and Karpathy' PG. Although time trends appear to correlate with performance, this is not a conclusive observation considering variations in time are not heavily weighted on reward, time and rewards are separate comparable factors to be measured given the 2:1 ratio of winning a match of Pong.

Furthermore, the implementation method used by Parthasarathy compared to Karpathy, could be the cause of a decrease in performance exhibited by Parthasarathy ILRCs.

2.7 FREQUENCIES

The frequencies of reward values acquired across 1,500 episodes initially do not reflect the true values of reward for the two PG implementations from Karpathy and Parthasarathy. When performing the result gathering stages, the environment issues a reward to four decimal places, the rewards are rounded to an integer value and so the frequency is not as accurate.

To elaborate, the reward sum in the first set of Karpathy' original solution is 73 but the true value after rounding to the nearest integer is189 and since the integers negate to accommodate for the difference from running reward, the frequencies do not reflect as accurately as they should. If they did factor in running rewards, what should be observed are trends shown in Figure 20. What may be observed in another 150 episodes is both trend lines beginning to remain or rise as the Agent learns to acquire more rewards in both solutions. Notably, the consistencies are also reflected.

The values in Table 13 exclude the second trial from Parthasarathy solution however, the results show all three ILRCs outperform the original solutions by an average of 5.4% and on average the ILRCs are accumulating reward 33.7% more. This shows the three ILRCs combined are performing better at accumulating rewards than the original implementations on average.

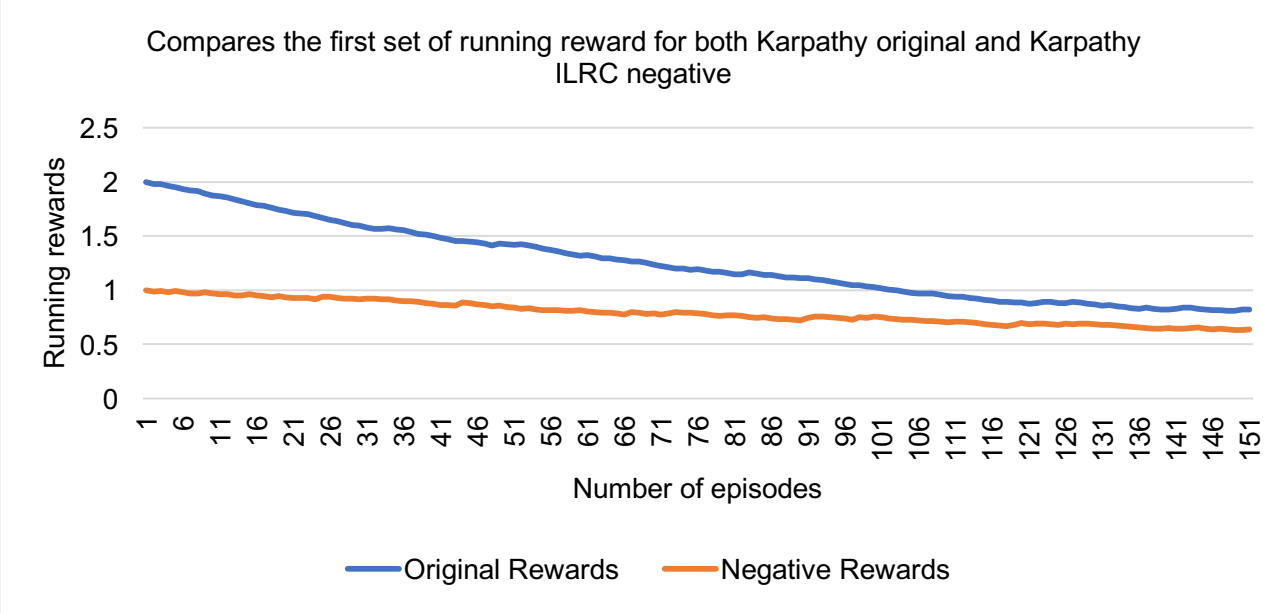


Figure 20 Compares Karpathy original to negative ILRC trends for reward accumulation, sampled from the first 150 episodes.

Developer	ILRC	1	2	3	4	5	6	Reward frequency %
Karpathy (PG)	Original	398	139	41	12	2	1	32.86
	Zero	427	146	38	7	3	0	41.4
	Negative	422	166	49	5	1	2	42.86
	Positive	404	170	31	13	1	0	41.2
Parthasarathy (PG Trial 1)	Original	433	134	40	13	0	0	41.33
	Zero	419	136	35	10	3	0	40.2
	Negative	404	139	44	3	1	0	39.4
	Positive	383	134	54	8	2	0	38.73
Parthasarathy (PG Trial 2)	Original	128	42	16	1	1	0	41.77
	Negative	118	35	18	6	0	0	39.33
Trial 2 Approximation	Original	426	140	53	3	3	0	41.66
	Negative	393	116	60	10	10	0	39.26

Table 11 Compares frequencies of reward values acquired across 1,500 episodes for each Agent.

Developer	ILRC	>= 135	!< 200	Reward frequency % >= 135	Reward frequency % !< 200
Frans (CEM)	Original	161	75	10.73	5
	Zero	153	73	10.2	4.86
	Negative	657	508	43.8	33.86
	Positive	80	18	5.33	1.2

Table 12 Compares the frequency of each episode successfully solving the CartPole environment with CEM across 1,500 episodes for each Agent.

ILRC	Mean Average of original %	Mean average for ILRC %	Difference %
Zero	28.31	30.6	2.5
Negative		42.02	13.71
Positive		28.42	0.11

Table 13 Compares the mean of reward accumulation frequency across three ILRCs to the mean reward accumulation frequency of three different RL solutions (CEM and two PG) for an average 1,500 episodes.

3 CONCLUSIONS

3.1 SUMMARY

Across the three different implementations in this study, results from logical combinations of reward values implemented in ILRC modifications, have presented an average 33.7% increase in reward accumulation frequency compared to the original solutions 28.3%. This is a 5.4% average improvement of reward frequency from Frans (2016a), Karpathy (2016a) and Parthasarathy (2016) with ILRCs of which it has been shown, CEM performs better than PG by a ratio of approximately 10:1 when reviewing the sum of rewards for inhibitors (negative ILRC).

The most efficient ILRC modification is negative (heavily penalising for negative actions) and outperforms the neutral and positive ILRCs by an average of 12.5% though, not in all implementations. In Parthasarathy' solution, positive ILRC (high rewards for positive actions) shows promising results. Looking forward, as the Agent starts to converge, a spike in performance may be exhibited which surpasses the original quite significantly as the Agent learns it is heavily rewarded when the positive actions it is learning, become more frequent.

It is important to remember, not all algorithms have been examined nor for longer periods of time during this study as such, these are preliminary results. Other potential values for immediate rewards have not been tested and none of these results have been examined when applied to robotics or online environments. Such studies will be able to confirm suitability for societal introduction however, they exhibit fluctuating yet promising statistics.

3.2 FUTURE WORK

Further exploration into the effects of ILRC modifications will be required for longer periods of time and against a larger variation of algorithm implementations and environment complexities. These studies hold the potential to yield larger quantities of results and will provide better statistics to gage a more conclusive result into the effects of ILRC on Agent behaviour.

It may be possible, through use of an additional NN or incorporating into a single NN, for an Agent to calculate an optimal immediate reward value for specific tasks and when encountering similar tasks, refer back and update the values accordingly synonymous to CEM. Although this seems unorthodox, humans cognitively assess their immediate success in solving a problem through the immediate reward and observations from the environment they are acting in, then use this experience in the future when confronted with similar problems (Newell & Simon 1972, p.148-149).

Word Count Including Tables: 4,519

4 REFERENCES

Brockman, G. and Christiano, P., 2016. *Concrete AI Safety Problems* [online]. France: OpenAI. Available from: https://OpenAI.com/blog/concrete-ai-safety-problems/ [Accessed 22 November 2016].

European Parliament, Committee on Legal Affairs, 2016. *DRAFT REPORT: With recommendations to the Commission on Civil Law Rules on Robotics* (2015/2103(INL)) [online]. Available from: https://goo.gl/b6qNF3 [Accessed 25 September 2016].

Frans, K., 2016a. *OpenAI-cartpole*. commit 8. [python script]. US: GitHub. Available from: https://github.com/kvfrans/OpenAI-cartpole

Frans, K., 2016b. *simple reinforcement learning methods to learn cartpole*. [online]. Panama, Central US. Kevin Frans. Available from: http://kvfrans.com/simple-algoritms-for-solving-cartpole/

IEEE, 2016. Artificial Intelligence & Ethics – Who does the thinking?: *IEEE AI & Ethics summit Report 2016*. [online]. Brussels, Europe: Available from: http://ieee-summit.org/wp-content/uploads/2017/01/IEEE_AI_SUMMIT_Report.pdf [Accessed 25 February 2017].

Karpathy, A., 2016a. *Training a Neural Network ATARI Pong agent with Policy Gradients from raw pixels*. 1. [Neural-Network training agent, Python]. Stanford: GitHub. Available from: https://gist.github.com/karpathy/a4166c7fe253700972fcbc77e4ea32c5

Karpathy, A., 2016b. Deep Reinforcement Learning: *Pong from Pixels* [online]. Available from: http://karpathy.github.io/2016/05/31/rl/ [Accessed 01 March 2017].

Newell, A. and Simon, A, H., 1972. *Human Problem Solving*. 104 (9) Englewood Cliffs, NJ: Prentice-Hall.

OpenAI, 2016. *CartPole-v0* [online]. OpenAI: Paris, France: Available from: https://gym.OpenAI.com/envs/CartPole-v0 [Accessed 26 April 2017].

Parthasarathy, D., 2016. *Neural Networks For Playing Pong*. commit 18 [RL Agent, Python] Dhruv Parthasarathy: Github, Available from: https://github.com/dhruvp/atari-pong [Accessed 13 March 2017].

Silver, D., 2015a. Lecture 1: *Introduction to Reinforcement Learning*: University College London: London. Available from: http://www.cs.ucl.ac.uk/sta/D.Silver/web/Teaching.html [Accessed 9 February 2017].

Silver, D., 2015b. Lecture 2: *Markov Decision Processes* [Video, online]. University College London: London: Deep Mind on YouTube. Available from: https://www.youtube.com/watch?v=lfHX2hHRMVQ [Accessed 14 February 2017].

UK Government, 2016. Data Science Ethical Framework: *Cabinet Office* [online]. London: Available from: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/524298/Data_science_ethics_framework_v1.0_for_publication__1_.pdf [Accessed 24 February 2017].

Wakefield. J., 2015. *Intelligent Machines: Do we really need to fear AI?* [online]. Available from: https://goo.gl/3evhlC [Accessed 24 September 2016].

BIBLIOGRAPHY

Abbeel, P. and Schulman, J., 2016. *Deep Reinforcement Learning through Policy Optimization* [online]. Available from: https://people.eecs.berkeley.edu/~pabbeel/nips-tutorial-policy-optimization-Schulman-Abbeel.pdf [Accessed 09 March 2017].

Altman, S., Brockman, G., Musk, E. and Sutskever, I., 2016. *OpenAI Technical Goals* [online]. France: OpenAI. Available from: https://OpenAI.com/blog/OpenAI-technical-goals/ [Accessed 22 November 2016].

Archbold, J. W., 1970. *Algebra*. 4th edition. London: Pitman.

Arbib, A. M. and Arbib, H. P., 2003. *The Handbook of Brain Theory and Neural Networks* [online]. 2nd edition. Cambridge, MA, U.S: The MIT Press.

Arisoy, E., Chen, S., Ramabhadran, B. and Sethy, A., 2015. Bidirectional recurrent neural networks for automatic speech recognition: *IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP) [online], 7179007, 19-24.

Barnard, E. and de Villiers, J., 2002. Backpropagation neural nets with one and two hidden layers: *IEEE Transactions on Neural Networks*. [online], 4(1), 136-141.

Barthelmess, U. and Furbach, U., 2014. *Do we need Asimov's laws?* 1st ed. [PDF ebook] New York: Cornell University. Available from: https://arxiv.org/pdf/1405.0961.pdf [Accessed 6 May 2017].

Barto, A. G. and Sutton, R. S., 1999. Reinforcement Learning: An Introduction: *Trends in Cognitive Sciences* [online], 3 (9), 360.

Bausel, B. R., 1986. *A Practical Guide to Conducting Empirical Research.* London, UK: Harper & Row Publishers Incorporated.

Bethard, S., Moens, M-F. and de Mulder, W., 2014. A survey on the application of recurrent neural networks to statistical language modelling: *Computer Speech & Language* [online], 30(1), 61-98.

Bournemouth University, 2016. *BU Guide to citation and referencing in the Harvard style*. Poole: Bournemouth University. Available from: http://libguides.bournemouth.ac.uk/bu-referencing-harvard-style/pdf-guide [Accessed November 2016].

Britz, D., 2016. *reinforcement-learning*. commit 155. [python script]. US: GitHub. Available from: https://github.com/dennybritz/reinforcement-learning

Brockman, G. and Christiano, P., 2016. *Concrete AI Safety Problems* [online]. France: OpenAI. Available from: https://OpenAI.com/blog/concrete-ai-safety-problems/ [Accessed 22 November 2016].

Brockman, G. and Schulman, J., 2016. *OpenAI Gym Beta* [online]. France: OpenAI. Available from: https://OpenAI.com/blog/OpenAI-gym-beta/ [Accessed 22 November 2016].

Cerf, G. V., 2013. What is a Robot?: *Communications of the Association for Computing Machinery* [online], 56(1), 7.

Clark, J., 2016. *CoastRunners 7* [Video, online]. YouTube. Available from: https://www.youtube.com/watch?v=tlOIHko8ySg&feature=youtu.be [Accessed 15 February 2017].

Cohen, P. R., 1995. *Empirical Methods for Artificial Intelligence*. 1st edition. Cambridge: The MIT Press.

Cox, D. D. and Dean, T., 2014. Neural Networks and Neuroscience-Inspired Computer Vision: *Current-Biology* [online]. 24 (18), 921-929.

de Campos, L. M. L., de Oliviera, C. L. R. and Roisenberg. M., 2016. *A Hybrid Neuro-Evolutive Algorithm for Neural Network Optimization* [online]. 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC 2016. Available from: http://ieeexplore.ieee.org/document/7727320/ [Accessed 01 April 2017].

Englander, I., 2003. *The architecture of computer hardware and systems software: An information technology approach*. 2nd edition. New York: John Wiley & Sons.

European Parliament, Committee on Legal Affairs, 2016. DRAFT REPORT: *With recommendations to the Commission on Civil Law Rules on Robotics* (2015/2103(INL)) [online]. Available from: https://goo.gl/b6qNF3 [Accessed 25 September 2016].

Frans, K., 2016a. *OpenAI-cartpole*. commit 8. [python script]. US: GitHub. Available from: https://github.com/kvfrans/OpenAI-cartpole

Frans, K., 2016b. *simple reinforcement learning methods to learn cartpole*. [online]. Panama, Central US. Kevin Frans. Available from: http://kvfrans.com/simple-algoritms-for-solving-cartpole/

Granger, E, B. and Pérez, F., 2007. IPython: *A System for Interactive Scientific Computing, Computing in Science and Engineering* [online]. 9 (3), p.21-29. Available from: https://ipython.org/ipython-doc/3/notebook/nbconvert.html

Graves, A., Hinton, G. and Mohamed, A-R., 2013. Speech recognition with deep recurrent neural networks: *IEEE International conference on Acoustics, Speech and Signal Processing* (ICASSP) [online], 13859735, 26-31.

Gunderson, P. J. and Gunderson, F. L., 2017. *Intelligence not equal to Autonomy not equal to Capability* [online]. PA, U.S: Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.78.8279&rep=rep1&type=pdf [Accessed 22 February 2017].

Hecht-Nielsen, R. and Nielson, R. H., 1990. Neurocomputing: *The technology of Non-Algorithmic information processing*. 5th edition. Reading, MA: Addison-Wesley Pub. Co.

Henderson, H., 2009. *Encyclopedia of Computer Science and Technology* [online]. 3rd edition. New York: Facts On File Incorporated.

Hughes, C. and Ilbury, R. H., 2014. *Qualitative and Quantitative approaches to social research* [online]. Warwickshire, UK: Available from: www2.warwick.ac.uk/fac/soc/sociology/staff/hughes/researchprocess/quantitative_and_qualitative_approaches.docx [Accessed 03 April 2017].

IEEE, 2016. Artificial Intelligence & Ethics – Who does the thinking?: *IEEE AI & Ethics summit Report 2016.* [online]. Brussels, Europe: Available from: http://ieee-summit.org/wp-content/uploads/2017/01/IEEE_AI_SUMMIT_Report.pdf [Accessed 25 February 2017].

Jain, P. L. C., Jain, L. C. and Martin, N. M., (eds.) 1998. Fusion of neural networks, fuzzy systems and genetic Algorithms: *IEEE Industrial applications*. Boca Raton, FL: CRC Press.

Jain, A. K., Mao, J. and Mohiuddin, M. K., 1996. Artificial neural networks: a tutorial: *Computer* [online], 29(3), 31-44.

Karpathy, A., 2016a. *Training a Neural Network ATARI Pong agent with Policy Gradients from raw pixels*. 1. [Neural-Network training agent, Python]. Stanford: GitHub. Available from: https://gist.github.com/karpathy/a4166c7fe253700972fcbc77e4ea32c5

Karpathy, A., 2016b. *Deep Reinforcement Learning: Pong from Pixels* [online]. Available from: http://karpathy.github.io/2016/05/31/rl/ [Accessed 01 March 2017].

Keuper, J. and Preundt, F-J., 2017. Distributed Training of Deep Neural Networks: *Theoretical and Practical Limits of Parallel Scalability* [online]. 2016 IEEE 2nd Workshop on Machine Learning in HPC Environments (MLHPC), Salt Lake City, UT 2016. Available from: http://ieeexplore.ieee.org/document/7835791 [Accessed 01 April 2017].

Korb, K. B., 2003. *Bayesian artificial intelligence*. United States: Chapman & Hall/CRC.

Le Blanc, C. and Stiller, E., 2002. *Project-Based Software Engineering: An Object-Orientated Approach.* London, UK: Addison-Wesley.

Maas, W., 1999. *Pulsed neural networks*. Edited by Wolfgang Maas and Christopher M. Bishop. 2nd edition. Cambridge, MA: MIT Press.

Mason, J., 1996. *Qualitive Researching*. London, UK: SAGE Publications Limited.

Miller, S., 2015. *A neural network library built in JavaScript.* 1. [Neural-Network Agent, JavaScript]. Segment: Github. Available from: https://github.com/stevenmiller888/mind

Min, K-S., Pham, V. K., Truong, N. S. and Yang, W., 2017. Memristor Circuits and Systems for Future Computing and Bio-inspired Information Processing: *IEEE Biomedical Circuits and Systems Conference (BioCAS) 2016* [online], 7833830, 456-459.

Nauck, D., Klawonn, F. and Kruse, R., 1997. *Foundations of neuro-fuzzy systems*. Chichester, England: John Wiley & Sons.

Nielsen, A. M., 2015. *Neural Networks and Deep Learning* [online]. Determination Press: Available from: http://neuralnetworksanddeeplearning.com/ [Accessed 14 February 2017].

OpenAI, 2016a. *OpenAI* [online]. OpenAI: Paris, France: Available from: https://OpenAI.com/ [Accessed 12 January 2017].

OpenAI, 2016b. *OpenAI Gym Core*.py. 2a0a2a3. [gym/core.py backend Script, Python]. OpenAI: GitHub, Available from: https://github.com/OpenAI/gym/blob/master/gym/core.py

OpenAI, 2016c. *Documentation* [online]. OpenAI: Paris, France: Available from: https://gym.OpenAI.com/docs [Accessed 12 January 2017].

Palmer, J., 2012. *Memristors in silicon promising for dense, fast memory* [online]. Science & Environment: BBC News, UK. Available from: http://www.bbc.co.uk/news/science-environment-18103772 [Accessed 01 April 2017].

Parthasarathy, D., 2016. *Neural Networks For Playing Pong*. commit 18 [RL Agent, Python] Dhruv Parthasarathy: Github, Available from: https://github.com/dhruvp/atari-pong [Accessed 13 March 2017].

Patyra, M. J. and Mlynek, D. M., (eds.) 1996. *Fuzzy logic: Implementations and applications*. Chichester, United Kingdom: John Wiley & Sons.

Penrose, R. and Gardner, M., 1999. *The emperor's new mind concerning computers, minds, and the laws of physics*. Oxford: Oxford University Press.

Python, 2017a. *Python 2.7.13 documentation* [online]. Oregon, U.S: Python.org. Available from: https://docs.python.org/2/ [Accessed 03 March 2017].

Python, 2017b. *15.3. time – Time access and conversions* [online]. Oregon, U.S: Python.org. Available from: https://docs.python.org/2/library/time.html?highlight=time#module-time [Accessed 04 March 2017].

Rossi, F., (ed.) 2013. IJCAI-13: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*: Beijing, China. (2 Vols). Palo Alto, CA, U.S.: AAAI Press.

Schalkoff, J, R., 1990. *Artificial Intelligence: An Engineering Approach.* New Jersey: McGraw Hill Publishing Company.

Sedwill, M. and Wallport, M., (eds) 2016. Artificial Intelligence: opportunities and implications for the future of decision making: *UK Government Office for Science:* Available from: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/566075/gs-16-19-artificial-intelligence-ai-report.pdf [Accessed 23 February 2017].

Shin, E., Subramanyam, G., Taha, M. T., Wang, S., Wang, W. and Yakopcic, C., 2017. Memristor Circuits for use in Neuromorphic Systems: *IEEE National Aerospace and Electronics Conference (NAECON) and Ohio Innovation Summit (OIS) 2016* [online], 7856807, 25-29.

Silver, D., 2015a. *Lecture 1: Introduction to Reinforcement Learning*: University College London: London. Available from: http://www.cs.ucl.ac.uk/sta/D.Silver/web/Teaching.html [Accessed 9 February 2017].

Silver, D., 2015b. *Lecture 2: Markov Decision Processes* [Video, online]. University College London: London: Deep Mind on YouTube. Available from: https://www.youtube.com/watch?v=lfHX2hHRMVQ [Accessed 14 February 2017].

Spears, W. M., 2000. *Evolutionary Algorithms: The role of mutation and recombination*. Germany: Springer-Verlag Berlin and Heidelberg GmbH & Co. K.

Sun, L., 2017. Existence of Periodic Solutions for a Discrete-Time Bidirectional Neural Networks: *IEEE Smart City and Systems Engineering (ICSCSE) International Conference* [online], 7825154, 25-26.

Tani, J. and Yamashita, Y., 2008. Emergence of Functional Hierarchy in a Multiple Timescale Neural Network Model: A Humanoid Robot Experiment: *PLOS one* [online], 4(11), e1000220.

Tsujii, J-I. and Shirai, Y., 1982. *Artificial Intelligence: Concepts, Techniques and Applications.* Chichester: John Wiley & Sons.

UK Government, 2016. Data Science Ethical Framework: *Cabinet Office* [online]. London: Available from: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/524298/Data_science_ethics_framework_v1.0_for_publication__1_.pdf [Accessed 24 February 2017].

Wakefield. J., 2015. *Intelligent Machines: Do we really need to fear AI?* [online]. Available from: https://goo.gl/3evhlC [Accessed 24 September 2016].

Warwick, K., (ed) 1991. *Applied Artificial Intelligence*. London: Peter Peregrinus Ltd.

Whitby, B., 2008. *Artificial intelligence: A beginner's guide*. UK: Oneworld.

Widrow, B., 1963. ADALINE: Smarter than sweet: *Stanford Today* [online]. Stanford, CA, U.S: Available from: http://www-isl.stanford.edu/~widrow/papers/j1963adalinesmarter.pdf [Accessed 18 February 2017].

ETHICS CHECKLIST

Bournemouth University

INITIAL RESEARCH ETHICS

NOTE: *ALL RESEARCHERS* MUST COMPLETE THIS BRIEF CHECKLIST TO IDENTIFY ANY ETHICAL ISSUES ASSOCIATED WITH THEIR RESEARCH. BEFORE COMPLETING, PLEASE REFER TO THE BU *RESEARCH ETHICS CODE OF PRACTICE* WHICH CAN BE FOUND WWW.BOURNEMOUTH.AC.UK/RESEARCHETHICS. SCHOOL RESEARCH ETHICS REPRESENTATIVES (OR SUPERVISORS IN THE CASE OF STUDENTS) CAN ADVISE ON APPROPRIATE PROFESSIONAL JUDGEMENT IN THIS REVIEW. A LIST OF REPRESENTATIVES CAN BE FOUND AT THE AFOREMENTIONED WEBPAGE.
SECTIONS 1-5 MUST BE COMPLETED BY THE RESEARCHER AND SECTION 6 BY SCHOOL

1 RESEARCHER DETAILS

NAME	RUSSELL ALLEN EAGLESFIELD CLARKE					
EMAIL	I7694005@BOURNEMOUTH.AC.UK					
STATUS	■ UNDERGRADUATE		☐ POSTGRADUATE		☐ STAFF	
SCHOOL	☐ BS	☐ AS	■ DEC	☐ HSC	☐ MS	☐ ST
DEGREE FRAMEWORK & PROGRAMME	BSC (HONS) COMPUTING					

2 PROJECT DETAILS

PROJECT TITLE	FOUR LAWS IN PRACTICE & NEUROLOGICAL NETWORKS.
PROJECT SUMMARY *SUFFICIENT DETAIL IS NEEDED; INCLUDE METHODOLOGY, SAMPLE, OUTCOMES ETC*	INVESTIGATION INTO THE BEHAVIOURAL NATURE OF AI SYSTEMS USING ALGEBRA WITH A PARTICULAR CONSIDERATION OF NEUROLOGICAL NETWORKS AND 'INTELLIGENT' SYSTEMS. INTENTION IS TO DERIVE SUFFICIENT RESULTS TO CONFIRM THE POSSIBILITY OF 'INTELLIGENCE' [AT A VERY BASIC LEVEL] AS OPPOSED TO AUTOMATION IN COMPUTING SYSTEMS. AT THIS STAGE IT IS NOT POSSIBLE TO MAKE A DECISION ON WHAT THE EVIDENCE OR MEASUREMENTS WILL BE AS FURTHER RESEARCH INTO THE STUDIES OF AI ARE REQUIRED HOWEVER, THERE WILL BE THROWAWAY PROTOTYPES AS SUCH, THE PROJECT MAY BE USING A HYBRID METHODOLOGY.
PROPOSED START & END DATES	SEPTEMBER 2016 – MAY 2017
PROJECT SUPERVISOR	MARCIN BUDKA MBUDKA@BOURNEMOUTH.AC.UK
FRAMEWORK PROJECT CO-ORDINATOR	HUSEYIN DOGAN HDOGAN@BOURNEMOUTH.AC.UK

3 ETHICS REVIEW CHECKLIST – PART A

I	IS APPROVAL FROM AN EXTERNAL RESEARCH ETHICS COMMITTEE (E.G. LOCAL RESEARCH ETHICS COMMITTEE (REC), NHS REC) REQUIRED/SOUGHT?	☐ YES	■ NO
II	IS THE RESEARCH SOLELY LITERATURE-BASED?	☐ YES	■ NO
III	DOES THE RESEARCH INVOLVE THE USE OF ANY DANGEROUS SUBSTANCES, INCLUDING RADIOACTIVE MATERIALS?	☐ YES	■ NO
IV	DOES THE RESEARCH INVOLVE THE USE OF ANY POTENTIALLY DANGEROUS EQUIPMENT?	☐ YES	■ NO
V	COULD CONFLICTS OF INTEREST ARISE BETWEEN THE SOURCE OF FUNDING AND THE POTENTIAL OUTCOMES OF THE RESEARCH? (SEE SECTION 8 OF BU RESEARCH ETHICS CODE OF PRACTICE).	☐ YES	■ NO
VI	IS IT LIKELY THAT THE RESEARCH WILL PUT ANY OF THE FOLLOWING AT RISK: LIVING CREATURES? STAKEHOLDERS?	☐ YES ☐ YES ☐ YES	■ NO ■ NO ■ NO

	Researchers?	☐ Yes	■ No
	Participants?	☐ Yes	■ No
	The environment?	☐ Yes	■ No
	The economy?		
VII	Does the research involve experimentation on any of the following: Animals?	☐ Yes	■ No
		☐ Yes	■ No
	Animal tissues?	☐ Yes	■ No
	Human tissues (including blood, fluid, skin, cell lines)?	☐ Yes	■ No
	Genetically modified organisms?		
VII 1	Will the research involve prolonged or repetitive testing, or the collection of audio, photographic or video materials?	☐ Yes	■ No
IX	Could the research induce psychological stress or anxiety, cause harm or have negative consequences for the participants or researcher (beyond the risks encountered in normal life)?	☐ Yes	■ No
X	Will the study involve discussion of sensitive topics (e.g. sexual activity, drug use, criminal activity)?	☐ Yes	■ No
XI	Will financial inducements be offered (other than reasonable expenses/ compensation for time)?	☐ Yes	■ No
XII	Will it be necessary for the participants to take part in the study without their knowledge / consent at the time?	☐ Yes	■ No
XII	Are there problems with the participant's right to remain anonymous?	☐ Yes	■ No
XIV	Does the research *specifically* involve participants who may be vulnerable?	☐ Yes	■ No
XV	Might the research involve participants who may lack the capacity to decide or to give informed consent to their involvement?	☐ Yes	■ No

4 Ethics Review Checklist – Part B

Please give a summary of the ethical issues and any action that will be taken to address these.

Ethical Issue: Social and Moral ethics to be considered and addressed in dissertation.	Action: To be addressed using intent of purpose.
• Defined in for and against • Justification • Speculation of potential future application for good of humanity	• Coding to help humanity as opposed to harm.

5 Researcher Statement

I believe the information I have given is correct. I have read and understood the BU Research Ethics Code of Practice, discussed relevant insurance issues, performed a health & safety evaluation/ risk assessment and discussed any issues/ concerns with a School Ethics Representative/ Supervisor. I understand that if any substantial changes are made to the research (including methodology, sample etc), then I must notify my School Research Ethics Representative/ Supervisor and may need to submit a revised Initial Research Ethics Checklist. By submitting this form electronically I am confirming the information is accurate to my best knowledge.

Signed	Mr. Russell. A. E. CLARKE	Date	24.09.2016

6 Affirmation by School Research Ethics Representative/ Supervisor

SATISFIED WITH THE ACCURACY OF THE RESEARCH PROJECT ETHICAL STATEMENT, I BELIEVE THAT THE APPROPRIATE ACTION IS:		
THE RESEARCH PROJECT PROCEEDS IN ITS PRESENT FORM	☒ YES	☐ NO
THE RESEARCH PROJECT PROPOSAL NEEDS FURTHER ASSESSMENT UNDER THE SCHOOL ETHICS PROCEDURE*	☐ YES	☐ NO
THE RESEARCH PROJECT NEEDS TO BE RETURNED TO THE APPLICANT FOR MODIFICATION PRIOR TO FURTHER ACTION*	☐ YES	☐ NO
* THE SCHOOL IS REMINDED THAT IT IS THEIR RESPONSIBILITY TO ENSURE THAT NO PROJECT PROCEEDS WITHOUT APPROPRIATE ASSESSMENT OF ETHICAL ISSUES. IN EXTREME CASES, THIS CAN REQUIRE PROCESSING BY THE SCHOOL OR UNIVERSITY'S RESEARCH ETHICS COMMITTEE OR BY RELEVANT EXTERNAL BODIES.		
REVIEWER SIGNATURE	Brelan	DATE 31/01/14
ADDITIONAL COMMENTS		

• Personally, in the ethics checklist, I think there should be a 'Potentially if research misused' column.

24.09.2016

DVD ROM CONTENTS