Investigating Comparisons Between Human Learning and Machine Learning Using the Dots and Boxes Game

EMMA DAVIS

Investigating Comparisons Between Human Learning and Machine Learning Using the Dots and Boxes Game

Student: Emma Davis

Supervisor: Dr. Ulrik Beierholm

MRes Cognitive Neuroscience

Department of Psychology, Durham University

2021

Investigating Comparisons Between Human Learning and Machine Learning Using the Dots and Boxes Game

Abstract

From its origins, machine learning has drawn inspiration from the human brain and its thought processes. Despite machine learning drawing so much from human learning, comparisons of the progression of learning between the two are fraught with difficulties. This study aims to explore the similarities and differences between human and machine learning, using a perfect information board game, Dots and Boxes, that both humans and machine learning agents learn to play by training against the same ‘box-greedy’ policy agent. Three q-learning reinforcement learning agents have been created with three distinct levels of strategy to compare against the learning progression and strategy of 62 human volunteers developed over 20 games each. Volunteers were also asked to complete BIS/BAS and CRT-MQC4 questionnaires after completing the Dots and Boxes task to study behavioural metrics that affect decision making, such as system 1 & 2 thinking and sensitivity to reward or punishment. Results from comparisons show how important context and prior knowledge is to human learning, and how machine learning agents can generalise better and reach optimal strategy faster with prior knowledge. Additional observations around how behavioural metrics affect individual decision making and correlations between machine learning agent strategy and human participant data allow for further comparisons to be made between human and machine learning.

Keywords: reinforcement learning; human learning; cognitive psychology; q-learning; perfect information board game.

Acknowledgements: Many thanks to Dr. Ulrik Beierholm for his support and guidance throughout this thesis, and many thanks to Nicole Li in providing support in completing the human participant experiments.

Contents

Introduction

2.1 Human Learning

The ability to learn is possessed by humans, animals and more recently by some machines. Human learning is essential to survival and quality of life, allowing individuals to acquire skills and achieve goals. G.A. Kimble defined learning as ‘a relatively permanent change in a behavioural potentiality that occurs as a result of reinforced practice’ (Hilgard, Kimble and Marquis, 1961). After a history studying human learning spanning hundreds of years, there are many learning theories and models used to explain human learning. While machine learning, where algorithms are trained to perform tasks, has drawn inspiration from human learning models, recent developments in machine leaning have begun to shed light on human learning in return. Examples of these recent developments include the work of Morse et al. (2015) discovering that our ability to retain new information partially depends on our physical relationship with it when studying children between 16-24 months old and a humanoid robot model, and the work of George Reeke showing that both humans and machines often need to perform a learning task themselves in order to understand its use in a different context when studying both human and AI systems while tracing letters.

There is no universal agreement on a single model of learning, rather multiple models for different aspects of learning; one of them being dual process theory. Dual process theory, suggested by Peter Wason and Jonathan Evans in 1974, proposes two distinct types of processes involved in human decision-making: heuristic processes, pertaining to the selection of relevant information, and analytic processes, pertaining to generating judgements or inferences on the selected information to ultimately make a decision (Wason and Evans, 1974). Daniel Kahneman developed dual process theory further through his own interpretation, separating the two processes into intuition, often called system one, and reasoning, often called system two (2003). While intuition allows for quick decision-making, judgement is influenced by strong emotional bonds and the reasoning process if biased by these. Reasoning, system two, is a slower and much more calculated process. In a study performed by Kannengiesser and Gero (2019), evidence for Kahneman’s systems 1 & 2 thinking can be observed when comparing the design protocols of professional engineers and engineering students, where it is shown the more experienced designers use more systems 1 thinking than students.

Often, situations that require humans to employ decision-making have a level of uncertainty involved regarding prospective wins and losses. In 1979, Kahneman and Tversky developed the behavioural model called prospect theory to explain how humans assess uncertainty. Prospect theory describes how individuals assess loss and gain perspectives asymmetrically, where people are much more sensitive to loss than to equivalent gains, also known as loss aversion. This bias affects an individual’s decision making in risky situations.

Building on cognitivism, which focuses on the study of mental processes as opposed to behaviour to study human learning, economist and cognitive psychologist Herbert Simon proposed the human decision-making theory that instead of behaviour being based on all available information being processed rationally to reach an ‘optimum’ decision, human decision making is about achieving outcomes that are ‘good enough’ based on limited available information and inability to process all information (Newell & Simon, 1973). Rim et al. (2012) display evidence of satisficing and maximising in human volunteers when completing various tasks involving gambling, binary choice and general decision-making competence. Maximising volunteers displayed a tendency to search for large amounts of information and thus had different information processing styles to satisficing volunteers. The concept of human decision-making being bounded by ‘cognitive limits’ adds another layer of complexity to human vs machine learning comparisons. Features well known in humans, such as constraints on information processing and memory, are not shared in machines. How computationally constrained humans can navigate and form strategy in complex tasks, and how machine learning can help our understanding of this, has been of great interest to cognitive psychology and machine learning scholars for many years.

One of the ways in which computationally constrained humans can make decisions in complex situations is through the use of satisficing. A concept first introduced by Herbert A. Simon (1956), satisficing is a decision-making strategy where decision makers can satisfice by finding optimum solutions for a simplified world when the true optimal solution is hard to find. The opposite of satisficing is maximising, where the truly optimal strategy in a given task or game is strived for. While machine learning algorithms are often maximising in their attempts to find the global optimum policy of a task, human decision making varies more widely on the satisficing-maximising scale with variables such as individual personality, complexity of task and abilities of the individual to consider.

While early machine learning aimed to replicate human learning to complete tasks, more recent studies have sought to draw comparisons between human learning and machine learning to help elucidate human learning mechanisms. Investigating the differences between human learners and machine learning algorithms when approaching the same task or facing the same opponent can offer a new perspective on human learning systems and allow us to improve our understanding of their underlying mechanisms.

2.2 Machine Learning

Despite the origins of artificial intelligence predating the 1950s, the term 'machine learning' was first coined in 1959 by Arthur Samuel, defining it as 'the field of study that gives computers the ability to learn without being explicitly programmed' (1959). Machine learning algorithms utilise various statistical methods and algorithms, which are trained to make classifications, predictions, complete tasks and even recognise patterns with higher accuracy than humans. The three main categories of machine learning algorithms are supervised, unsupervised and reinforcement learning algorithms (Jordan & Mitchell, 2015). In supervised machine learning, data with a clearly defined output is fed into an algorithm and feedback is administered depending on how well said algorithm can learn the data and apply its understanding to make predictions or classify unseen data. Contrary to this, unsupervised learning does not use data with a clearly defined output; instead, an algorithm learns patterns and structures in the data it is fed. Reinforcement Learning is a type of machine learning technique that enables an agent to learn in an interactive environment by trial and error using feedback from its own actions and experiences. Reinforcement learning differs from supervised learning in not needing labelled input/output data, and differs from unsupervised learning which assumes no actions or consequences; instead, learning is driven by a reward/punishment system that provides feedback to agents. Reinforcement learning is the closest machine learning paradigm to typical human learning, where sub-optimal actions are not explicitly corrected and instead outcomes of a series of actions are observed, similar to how, in most situations, children learn via feedback from exploration of their environment instead of from a parent constantly providing feedback with each action. In reinforcement learning there is emphasis on finding a balance between exploration of unknown environment and exploitation of current knowledge (Kaelbling, Littman and Moore, 1996). For the purposes of this paper, focus will largely be on reinforcement learning (RL).

Most machine learning algorithms can be broken down into three components, each of which are run iteratively as the algorithm trains until appropriate accuracy is achieved:

- Decision Process: calculations that take the input data and predict the output.
- Error Function: a method measuring how good the prediction was by comparing to known examples if they are available (this is less applicable to unsupervised learning).

- Optimisation Process: a step where, using the error function, the decision process is altered to make better predictions and reduce error.

The 1990s saw a shift in machine learning away from small-scale knowledge-based models towards ones driven by data. The need for scalability to accommodate large datasets, alongside ability to automate extraction of features in data instead of manual handpicking, led to the advent of deep learning by the 2000s. A sub-field of machine learning, deep learning algorithms largely follow the same process but leverage artificial neural networks with multiple layers to extract features from large datasets to aid predictions. Input data is transformed into abstract, decomposed representations at each neural network layer to extract features, similar to feature extraction in visual cortices of mammals. How individual brain cells conveyed information in the brain for feature detection, first discovered by Hubel and Wiesel (1962), inspired the development of the Neocognitron (Fukushima, 1980), which in turn inspired the production of further neural networks. The 'all-or-none' characteristic of biological neurons, where a neuron either generates an action potential or remains silent depending on its given input, is mimicked in artificial neural networks, simplifying the chemical and electrical activity in synapses as connected layers of numeric matrices. Through the training of this simplified network of artificial neurons with vast amounts of input data, effective feature extraction can be achieved.

2.3 Human and Machine Learning Comparisons

From Alan Turing's 'Turing Test' (1950), where in order to pass a computer must be able to fool a human into believing it is also human, to Frank Rosenblatt's perceptron, designed to simulate the thought processes in the human brain (1958), to more recent machine learning advancements, machine learning has been influenced by human learning and intelligence since its origins. However, despite machine learning drawing so much from human learning and both human and machine learning being experience-driven, it would be naïve to suggest the two are completely equivalent. Neural networks are merely imitations of human brains, limited by our lack of understanding and inability to recreate such complex nervous systems in a computational domain (Zador, 2019). Further distinctions have been made between human and machine learning in countless studies; Dubey et al show the importance of prior knowledge when humans learn a new task (2018), allowing transfer of knowledge across different domains or problems. Conversely, most machine learning algorithms often start learning from scratch and as such don't have any prior knowledge reserves or context to draw from, leading models to become very specialised in completing one particular task but unable to effectively transfer that learning to another domain. Additionally, humans are able to learn from little experience, whereas machine learning algorithms typically need large amounts of data in order to effectively complete a task. In a pattern identification study, carried out by Kuhl et al (2020), comparing human and machine learning performance on the same amount of training data the performance of machine learning algorithms varies wildly across different patterns while human volunteers display more consistent performance. Furthermore, human performance plateaus after 20 instances whereas machine learning is much slower to plateau. Wang et al show faster learning in humans as a result of meta-learning in the prefrontal cortex, modelled as a recurrent neural network with weight adjustments driven by dopamine release (2018). Through this meta-RL system, humans may transfer prior learning to the learning of new tasks, thus speeding up learning.

While there are many differences between human and machine learning, the learning process applied to train machine learning algorithms draws from some of the core principles behind human learning. Humans learn and acquire knowledge through experience, either directly or via the experience of others. Similarly, machines learn and acquire knowledge through experience in the form of data. A combination of consolidating information to memory and ability to transform memory to knowledge and skills is required for humans and machines alike to learn how to solve problems. A human may memorize the solutions to a set of problems, but if they fail to transform that memory to problem solving skills then when posed with a new problem they will likely fail to solve it. Likewise, in machine learning all data may be memorised by a model, referred to as overfitting, leading the model to lack any ability to generalise and apply its learning to a new problem.

Reinforcement learning algorithms' origins lie in trial-and-error learning observed in animals by psychologists such as Thorndike. According to Thorndike, when no clear solution to a problem or a task is known to the learner they adopt a trial-and-error method of trying solutions and rejecting them if they do not bring the learner closer to completing the problem. Through this process the learner can eventually adopt a strategy formed from actions that lead to a solution to the problem (Thorndike, 1898).

Similar to trial-and-error learning, reinforcement learning algorithms aim to learn a strategy, often referred to as a policy, by interacting with the environment or task and observing the result of their actions in the form of positive or negative rewards. The basic framework of RL draws from Markov Decision Processes (MDP), discrete-time stochastic control processes that model decision making processes.

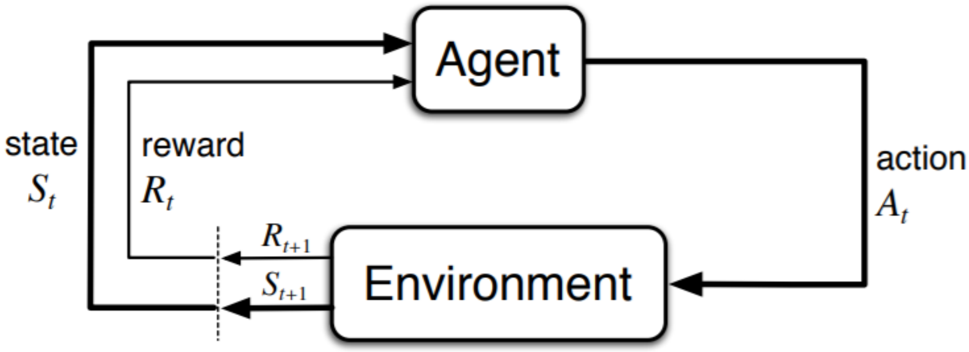


Figure 1: Interaction between an agent and environment in a Markov Decision Process (Sutton & Barto, 2018).

A Markov decision process is a 4-tuple (S, A_s, P_a, R_a), illustrated in Figure 1, where:

- S is a set of states, known as the state space
- A_s is a set of actions available from state S, known as the action space
- $P_a(s, s')$ is the probability that performing action a from state s at time t will lead to state s' at time $t + 1$
- $R_a(s, s')$ is the immediate reward received after transitioning from state s to state s' via action a

The goal in MDP is to find an optimal policy: a function π that specifies the action $\pi(s)$ that the algorithm, referred to as an agent, should choose in state s. This optimal policy can be found through applying q-learning, first introduced by Chris Watkins in his PhD thesis as a dynamic programming solution to policy finding in Markov Decision Processes (1989). Q-learning is a model-free reinforcement learning algorithm that learns the value of an action from a given state, and can ultimately find an optimal policy in any finite Markov Decision Process by maximizing the value function. Model-free, in the context of q-learning, refers to whether an agent must predict the response of the environment; in other words an algorithm that estimates the optimal policy without estimating the dynamics (transition and reward functions) of the environment. Model-based agents learn a model of their environment from observations and plan an optimal solution using said model, thus estimating dynamics such as transition function and reward function with $p(s' , r|s, a)$. As such, model-based learning lends itself to stochastic games or tasks with probabilistic transitions and model-free learning lends itself to deterministic games.

Determining the quality of a state-action combination can be summarised in the function:

$$Q : S \times A \rightarrow R$$

Where Q is a numeric matrix, the size of which is determined by the total number of states and actions. At the start of learning values in Q are set to an aribtrary number, then for each time point t where the agent is in state s the agent selects an action a_t, observes reward r_t, enters a new state s_{t+1} and the corresponding value in Q is updated.

At the core of q-learning is the Bellman equation, used to update values in the Q matrix via a weighted average of old information and new information gained through the agent's trial and error. At each time step, values in Q are updated with the function:

$$Q^{new}(s_t, a_t) \leftarrow Q^{old}(s_t, a_t) + \alpha \cdot \left(r_t + \gamma \cdot maxQ(s_{t+1}, a) - Q^{old}(s_t, a_t)\right)$$

Updating $Q^{new}(s_t, a_t)$ can be broken down into three factors:

- $\alpha \cdot r_t$: reward gained when action a_t taken in state s_t, weighted by the learning rate
- $\alpha \cdot \gamma \cdot maxQ(s_{t+1}, a)$: the maximum reward that can be gained from state s_{t+1}, weighted by the learning rate and discount factor
- $(1 - \alpha)Q(s_t, a_t)$: the current value of the state – action pair, weighted by the learning rate.

The learning rate $(0 < \alpha \leq 1)$ is a tuning parameter that determines each iterations step size and influences the amount newly acquired information overrides old information. A higher learning rate means Q 'learns' faster, as old information is overridden at a higher magnitude. The discount factor $(0 < \gamma \leq 1)$ is also a tuning parameter that determines how much an agent prioritises future rewards over immediate rewards.

Q-learning belongs to the temporal difference (TD) learning class of reinforcement learning models, where an agent aims to update state-action values with the predicted total rewards over time in mind. By bootstrapping the maximum estimated next state-action values and incorporating them in the current timestep state-action q-value estimate, the agent can learn faster through the comparison of consecutive values and can develop a 'lookahead' policy that prioritises larger long-term rewards.

Temporal difference learning has received a lot of interest by cognitive psychology and neuroscience due to similarities between error functions in these algorithms and neural 'error functions' in the brain. 'A Neural Substrate of Prediction and Reward', one of the first studies to draw these comparisons, compares the TD algorithm to the role played by dopamine neurons when predicting future rewards where the firing of dopamine neurons encodes reward prediction error (Schultz, Dayan & Montague, 1997). This study highlights data produced from trials where monkeys touch a lever after the appearance of a small light, after which a reward may or may not be delivered. In trials where a reward is not delivered after onset of the light, dopamine neurons are observed to fire below their basal rate. Additionally, before and in the early stages of training most dopamine neurons show a short burst of firing activity after reward delivery, which slowly changes to bursts of firing as soon as the light is illuminated after several days of training. These data combined suggest expected reward delivery based on the occurrence of light is encoded into the fluctuations of dopaminergic activity, encoding what we know to be reward prediction error.

Additionally, Seymour et al. (2004) used fMRI to study brain responses of fourteen healthy human subjects during a second-order pain learning task and observed neural activity in the ventral striatum and anterior insula corresponding to the signals for sequential learning predicted by temporal difference models. Volunteers were shown two visual cues after which a low or high intensity pain stimulus was delivered, where the second visual cue was fully predictive of the pain stimulus and the fist visual cue was probabilistically predictive of pain stimulus. As volunteers progressed through trials they learned the correct associations between stimulus, allowing for the study of not only expectation of pain but also the reversal of expectation. Trial data shows temporal difference prediction error slowly decreasing in later cues and increasing in the first cue as volunteers were able to make pain predictions. fMRI data of volunteer brains displayed significant correlation between specific brain regions and the prediction error, the signal driving learning, thus revealing a flexible aversive learning process in human learning. Among the highest correlations (statistically significant post-correction for multiple comparisons) are the right and left ventral putamen correlating with temporal difference prediction error with z-scores of $z = 5.38$ and $z = 3.93$ respectively, and the right head of caudate correlating with temporal difference prediction error with a z-score of $z = 3.75$.

Deep reinforcement learning is a subfield of machine learning that combines reinforcement learning, such as q-learning, and deep learning. Successful applications of reinforcement learning using neural networks such as TD-Gammon (Tesauro, 1995) and the first practical demonstration of backpropagation (LeCun et al., 1989) pioneered the advancement of deep reinforcement learning, leading to the development of the Deep Q-Network algorithm by DeepMind in 2015 (Mnih, Kavukcuoglu & Silver, 2015) to address the need for wider application of reinforcement learning outside of fully observed low-dimension state spaces, or where useful features for learning cannot be manually selected. Since these early advancements further game-oriented reinforcement learning agents have been developed for increasingly complex tasks. AlphaGo was able to defeat the human European Go champion using deep neural networks with tree search and training with simulated self-play (Silver, Huang, Maddison, et al. 2016).

Such as in reinforcement learning, deep learning agents learn to make decisions by trial and error, however this learning process is aided by neural networks approximating expected reward from state-action pairs. In Deep Q-learning, a neural network is used to approximate the q-value function, where representations of game states are fed in as input and q-values of all possible actions are generated as output. There are a number of neural network architectures, with Convolutional Neural Networks (CNNs) being among the most popular. Inspired by the neuronal organization in the animal visual cortex, CNNs are multilayer perceptrons regularized so that local connectivity can be used to learn patterns instead of full connectivity between neurons, similar to cortical neurons responding to stimuli in restricted regions of the visual field (Fukushima, 1980). The relatively sparse connectivity between layers helps reduce risk of overfitting, where models learn training data too well and fail to generalise unseen data, and the use of kernels panned across input data means features can be learned and extracted irrespective of their location in data.

2.4 Dots and Boxes

Board games make for an effective paradigm to study both human and machine learning, acting as complex but contained environments for either humans or machines to showcase learned strategy. Samuel's ground-breaking work applying machine learning to the game of checkers (1959) paved the way for further board game algorithms, such as Quinlan's application of decision trees to chess end-games (1983) and the application of search algorithms and board pattern recognition to Othello (Lee & Mahajan, 1990). Board games with perfect information, where players have the same information that would be available at the end of the same (i.e. Chess, Tic-Tac-Toe or Go), are useful mediums to study human and machine learning for many reasons. Perfect information games are woven into the history of AI and its advancement; between the 1990s and 2000s many landmark goals of AI were achieved, including the defeat of chess grand master Gary Kasparov by IBM's Deep Blue (Campbell, Hoane and Hsu, 2002). Practically, perfect information games can range in complexity and the lack of stochastic processes allows learners to develop and apply intelligent strategy without the setbacks of chance.

Dots and Boxes is a perfect information deterministic two-player game created by French mathematician Edouard Lucas in the 19th century. Players start with an empty grid of dots and take turns to place single lines, either horizontally or vertically, between two unconnected adjacent dots. When a player completes the fourth side of a 1 x 1 box, they earn one point and must take another turn. The player with the most points (i.e. most boxes claimed) once no more lines can be placed wins. Boards may be any size larger than 2x2 boxes (3x3 dots) and don't necessarily have to be symmetrical, but for the purposes of this study this paper focuses on 3x3 box grids only.

Optimal Dots and Boxes strategy varies depending on board size, however there are general stages to the development of strategy that may be applied to most boards. Weaver and Bossomaier (1998) distinguish several levels of strategy that can be applied against an opponent on a 3x3 box grid. The first phase of strategy is classed as random, where players place lines arbitrarily with no lookahead or examination of the board. Players do not register opportunities for claiming boxes nor avoid allowing their opponent to claim boxes.

The second phase of strategy is classed as greedy, where players will claim a box if the opportunity is made available. The third phase of strategy builds on the second greedy phase, but with the addition of the player applying one step look ahead to avoid giving their opponent the opportunity to claim a box by avoiding placing third sides of boxes if they are able.

The fourth phase of strategy is reached when a player can learn to optimally concede boxes for a larger future reward. The fifth phase of strategy builds on the fourth phase with the addition of the player avoiding completing some squares. A player may want to avoid completing squares if their opponent is baiting them into completing short chains, as seen in Figure 2.

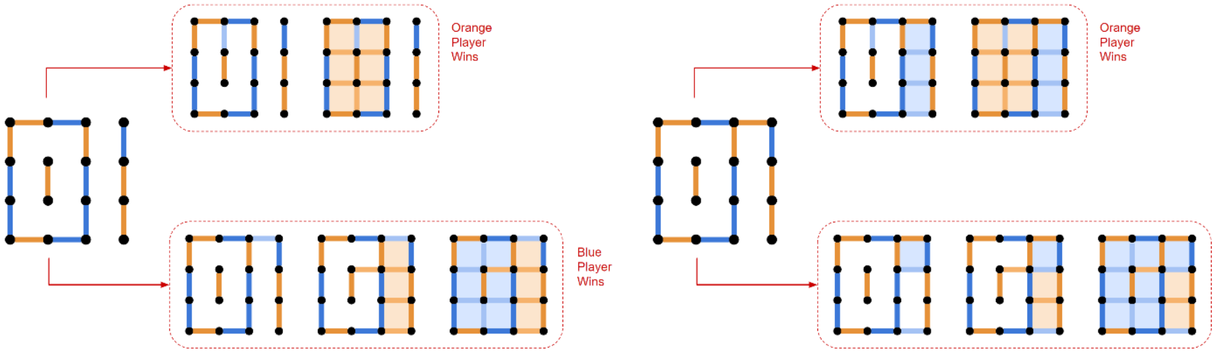


Figure 2: Examples of the fourth and fifth levels of strategy described. The left image displays an example of the fourth phase of strategy, where player blue can choose to bait the opponent (orange) into claiming the shorter chain and consequently opening up the longer chain for blue to claim. The right image displays an example of the fifth phase of strategy, where player blue can avoid claiming the last two boxes of the short chain and bait the opponent into claiming them instead, consequently opening up the longer chain for blue to claim.

Both phases four and five of strategy require a certain amount of 'unlearning' greedy strategy, which can pose a problem both in human and machine learning, where agents must be adaptable depending on the opponent faced. Additionally, the temporal strategy required in phases four and five require players to avoid using impulsive system 1 thinking and rely on system 2 reasoning instead. Weaver and Bossomaier's heuristic network was only able to reach the first phase of strategy defined, and while more recent attempts to model Dots and Boxes agents have seen higher win rates very few studies analyse the resulting agent strategy. As such, it is still an open question to what degree machine learning agents are able to achieve these levels of strategy or higher, and whether the learning trajectory of said agents follow that of human learners.

Inhibition control can vary across individuals and may be measured in multiple ways, including via the BIS/BAS scale. The BIS/BAS scale is a self-report questionnaire of 24 questions rated on a 4-point Likert scale (Carver & White, 1994) with questions are designed to measure two motivational systems, the first being the Behavioural Inhibition System (BIS) and the second being the Behavioural Activation System (BAS). The questionnaire yields one BIS score and three BAS scores, defined as:

- BIS: Sensitivity to punishment, motivation to avoid aversive outcomes.
- BAS Drive: Motivation pertaining to the persistent pursuit of desired goals.
- BAS Fun Seeking: Motivation pertaining to a desire for new rewards and willingness to approach a potentially rewarding experience on impulse.
- BAS Reward Responsiveness: Sensitivity to positive outcomes and positive attitude towards rewards.

Many behavioural theorists believe these two motivational systems underlie behaviour and ability to control impulse.

The Cognitive Reflection Test (CRT-MCQ 4) is a multiple-choice test designed to measure intuition, inhibition and cognitive reflection. The test reliably predicts reasoning performance and decision-making. Volunteers were offered a seven question, four-option response format CRT to complete after playing Dots and Boxes. Each of the seven questions has one correct answer, one intuitive (incorrect) answer and two other incorrect answers. Often, participants respond to questions with an intuitive, yet ultimately wrong, answer. For a participant to get the correct answer, they must take some time and apply analytical processing via system two reasoning, instead of relying on system one intuition alone (Sirota & Juanchich, 2018).

The prefrontal cortex controls many of the executive functions of the brain involved in decision making and learning, such as planning, reasoning and impulse control. Impulsive individuals make risky decisions, choosing immediate rewards despite potential long-term negative consequences (Moeller et al., 2001). As such, when studying learning and decision making in humans it is important to assess individual risk appetite and impulsivity. The Barratt Impulsivity Scale (BIS 11) is a questionnaire designed to assess impulsiveness in individuals (Patton et al., 1995); however for the purposes of this study BIS/BAS and CRT-MCQ 4 were deemed sufficient to infer impulsivity as well as other behavioural characteristics. With the combination of BIS/BAS self-reporting, where impulsive behaviour may be seen as a joint function of BAS Fun and BAS Drive in particular (Poythress et al., 2008), and CRT-MCQ, which directly assesses impulsivity over reasoning, impulse control of individuals can be evaluated.

Despite being a perfect information and deterministic task, and thus lending itself to reinforcement learning, Dots and Boxes has been little studied as a reinforcement learning task compared to the likes of Chess and Go. While there are few studies to reference, those that have been conducted have trialled various reinforcement learning algorithms to varying degrees of success. Bossomaier & Knittel (2006) apply an artificial economics model to the Dots and Boxes task, where policy is learned as a series of behavioural rules instead of through q-learning. However, the artificial economics model required hundreds of thousands of games to train against, and performance against a sophisticated artificial player did not exceed a 40% win rate. Zhang, Li & Xiong (2019) compare a Monte-Carlo tree search Dots and Boxes agent against AlphaZero, a general-purpose deep learning algorithm that has mastered games such as Shogi, Chess and Go (Silver et al., 2018), and found the Monte-Carlo algorithm failed to reach the same levels of success as AlphaZero.

As Dots and Boxes is relatively unknown compared to other board games, and with strategies of varying complexity and a multitude of ways reinforcement learning agents can be trained to complete it, it is therefore a perfect task to use to study both human and machine learning. In this study multiple q-learning agents were developed to capture different learning progressions and levels of strategy for comparison with human volunteer data. To draw comparisons between machine learning and human learning, both human participants and machine learning agents 'trained' against a box-greedy Dots and Boxes q-learning agent, after which data such as board states, actions and wins/losses were collected for analysis.

Methodology

With the difficulties faced modelling Dots and Boxes via Monte-Carlo (Zhang, Li & Xiong, 2019) and economics models (Bossomaier & Knittel, 2006) in mind, q-learning was selected as the most appropriate algorithm to model the Dots and Boxes task due to its successes when applied to similar board game tasks such as Connect 4 (Alderton, Wopat & Koffman, 2019). In this study, three q-learning models were developed; one q-learning with linear approximation of features model and two Deep-Q learning models with the same architecture but trained in different ways. The resulting 'box-greedy' q-learning with linear approximation of features agent was used as an opponent for human volunteers to play 20 games against, as well as an opponent for the two Deep-Q models to train against.

3.1 Q-Learning with Linear Approximation of Features

3.1.1 Architecture

To effectively model different learning progressions and different stages of strategy, multiple models utilising different reinforcement learning techniques were needed. While the first stage of strategy defined by Weaver and Bossomaier could be captured effectively via a random policy agent, more advanced strategy required q-learning to train agents to develop more effective policies. A basic q-learning agent for a 4x4 dot board was first created - however the Q-value table of all possible actions, of which there are 24 on a 4x4 dot board, against all possible board states, of which there are 2^{24} = ~17 million on a 4x4 board, was too large and cumbersome to train effectively. State-space had to be approximated to reduce the size of the Q-value table to a trainable size.

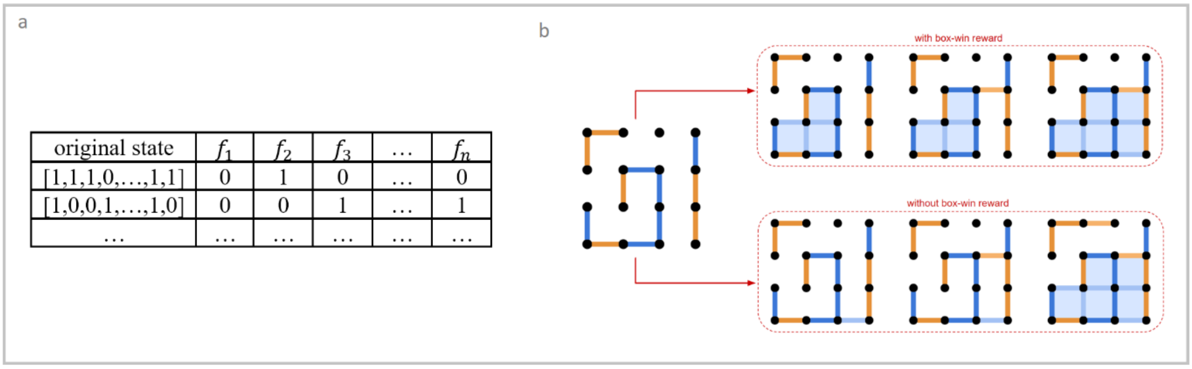
The best method for approximating state-space largely depends on the qualities of the state-action space and whether tasks are linear or non-linear. Approximating states as a sum of the state-space features is a simple but effective function approximation method, used here as an adaptation of the method outlined by Melo & Ribeiro (2007) and adapted for the Dot and Boxes task. Q-learning with linear combination of features provides a solution to large state spaces where states can be represented by a sum of their features, meaning the value of state-action pairs can be represented as so:

$$Q(s,a) = w_0 + w_1 f_1(s,a) + w_2 f_2(s,a) + \cdots + w_n f_n(s,a)$$

The aim is to use sets of handcrafted features to generalise the estimation of state-action pairs with similar features. This turns updating the Q-value matrix into a regression problem, where the goal is to learn the function mapping state-action features to their state-action Q-values (i.e. $f(s,a) \rightarrow Q(s,a)$). Instead of using Bellman update steps to update Q-values directly, the Bellman equation is used to update feature weights $w = (w_0,\, w_1, \dots, w_n)$ through gradient descent, as so:

1. Weights $w = (w_0, w_1, \dots, w_n)$ are initialised randomly
2. For each weight update step, do:
    a. Observe reward r and next state s' from state s and action a.
    b. Update weight via $w \leftarrow w + \alpha \cdot (r + \gamma \cdot maxQ(s', a) - Q(s, a)) \cdot \overrightarrow{\nabla_w} Q(s, a)$, where $\overrightarrow{\nabla_w} Q(s, a)$ is the gradient of $Q(s, a)$ in w.
3. Move from state s to s' and repeat.

With Bellman update steps being performed to update weights w_i in place of a large Q-value table, this function approximation reduces the state space size from $|S|$ to |F| (where F is the domain for $f \in F$). w_0 is the bias term weight of $f_0(s, a) = 1$, which is used as a universal feature for $f \in F$. The gradient $\overrightarrow{\nabla_w} Q(s, a)$ provides the direction in which the weight updates are performed, allowing weights to converge to the optimal values and thus optimal Q-values.



original state	f_1	f_2	f_3	…	f_n
[1,1,1,0,…,1,1]	0	1	0	…	0
[1,0,0,1,…,1,0]	0	0	1	…	1
…	…	…	…	…	…

*Figure 3: **a** Example of an original state (within domain of 2^{24} state space) represented as features within a domain of 2^n state space, where n < 24. **b** Example training games from q-learning linear combination of features agents, one with rewards granted for box-wins as well as overall game wins and one with rewards only given for overall game wins. The agent that was only rewarded for overall game wins, while ending up with the same number of boxes won as the box-win reward q-learning agent, would often employ counterintuitive strategy. As the random policy agent opponent was unlikely to claim boxes at random in early stages, this behaviour would not be penalised.*

Function approximation, while effectively reducing state space size and thus increasing the speed at which an optimal policy is found, does have many limitations. When characterising states in terms of their features, selection of features can be limited by the understanding of the game environment and strategy of those choosing the features. Unsupervised feature learning methods such as autoencoders or independent component analysis may be applied in these situations, but for the purposes of this model manual feature engineering was appropriate. Features selected for this model were largely handpicked and aimed to characterise the 24-edge board in terms of the 9 boxes available to claim instead, with features such as whether a state-action pair claims a box or sacrifices a box (i.e. allows the opponent to claim said box).

While effective at reducing training time and model complexity, reducing states to their features is often lossy and if chosen features fail to characterise an element of a state integral to the optimal policy, then the optimal policy for the original state space cannot be found; an optimal policy for the function approximated state space can only be found instead. Though these limitations are not ideal in the pursuit of finding the ultimate optimal strategy, it provides an interesting agent to compare against human strategy, which is often itself restricted by cognitive limitations. The concept of satisficing, where sub-optimal 'good enough' solutions are found for simplified versions of tasks, is often used in human decision making. This concept has since been introduced into machine learning studies where satisficing can be used to find a strategy that fits the ‘aspiration level’ much faster than finding the optimal strategy (Tamatsukuri & Takahashi, 2019).

3.1.2 Training

Learning rate α and discount factor γ were both selected in line with existing board game q-learning agents and advice from Even-Dar & Mansour (2003), which suggests the optimal learning rate for a random policy Markov Decision Process is $\alpha = \frac{1}{t^{\omega}}$, $\omega = 0.85$. Generally, the discount factor is high and often arbitrarily set to $\gamma = 0.9$, resulting in q-values better representing cumulative future reward. This is ideal when finding the optimal policy for a task that has a temporal nature, such as Dots and Boxes. It is important to find a balance between valuing short-term vs long-term reward in Dots and Boxes as while long-term reward strategies that conceed boxes are crucial to winning against strong opponents, it is also important not to miss out of greedy opportunities early on in the game while the structural ‘traps’ (such as open box chains) needed for long-term plays are not yet present on the board.

Hyperparameters set as $\alpha_{init} = 0.85$ and $\gamma = 0.9$ provided stable training and efficient learning in q-learning with linear combination of features, with learning rate decaying over the first $N = 2000$ training games to $\alpha_{end} = 0.25$ as follows:

$$r = max\left(\frac{N - n_{step}}{N}, 0\right)$$

$$\alpha \leftarrow (\alpha_{init} - \alpha_{end})r + \alpha_{end}$$

The q-learning with linear combination of features agent was trained against a random policy agent over multiple games until the q-learning agent’s win rate plateaued. The weights of approximated feature-action pairs were updated at the end of each game, and resulting q-values were stored as a numeric matrix upon completion of training. Rewards were granted to all state-action pairs in a game if the q-learning agent won, and all state-action pairs in a losing game were penalised. Any state-action pairs that directly led to q-learning agent completion of a box were also rewarded a small number of points. This was necessary in order to accelerate training and avoid the q-learning agent developing a policy that encouraged acting randomly in early game states, which was behaviour seen in the initial training trials but not observed once implementing box completion rewards (Fig 3b).

3.2 Deep Q-Learning

3.2.1 Architecture

Due to limitations such as lossy state representations, q-learning with linear approximation of features policy was only able to converge to a ‘box greedy’ strategy; a more optimal policy required another state-space approximation approach. As such, two Deep Q-learning agents were developed and trained.

When deciding on Deep Q-learning neural network architecture, the kind of neural network that fits the problem best is often down to the kind of data an agent is learning from and the qualities of the game or task environment. Neural networks of fully-connected (or dense) layers, while structure agnostic and applicable to most input types, often have much weaker performance than purpose-built architectures. Unlike other hidden layers such as convolutional layers, fully-connected layers often struggle to learn features and as such can fail to generalise board states not seen before in training but containing features observed in previous training data.

When dealing with learning sequential games or tasks, Long Short-Term Memory (LSTM) models are often used due to their ability to remember values over arbitrary timesteps. However, learning a strategy in environments with Markov properties, such as Chess or Dots and Boxes, is less reliant on sequential board states and more reliant on the spatial relationships and board ‘features’ of a single board state. As such, architectures such as Convolutional Neural Networks (CNN) are better suited to working with matrix representations of board states, and are able to develop internal representations of the board states to learn spatial relationships and patterns.

CNNs consist of an input layer, hidden layers and an output layer, where hidden layers include layers that perform convolutions finding the dot product between kernels and the layer’s input matrix. As the smaller kernel matrix filters over the input matrix in a layer, a feature map is generated which contributes to the input of the next layer. Pooling layers are then used to down sample the feature maps, summarising the features generated in the convolution layer. Before the output layer, flattened feature maps from the final pooling layer are fed into a fully connected layer where non-linear combinations of features are found and then fed to the output layer. Finally, the output layer returns classifications or predictions based on the data fed into the input layer. Between hidden layers, activation functions are used as a threshold to decide if data should be fed on to the next layer’s neurons. Rectified Linear Activation Unit (ReLU) is regarded as the best activation function in most deep learning situations, defined as $g(z) = max\{0, z\}$. The ability to output true zero values introduces sparsity in the following layers and speeds up the learning of features.

In the Deep-Q architecture developed for this study, hidden layers comprised of three lots of convolution layers with ReLU, each followed by max pooling layers. After three convolutions, flattened feature maps are fed into a dense fully-connected layer which reshapes output to a flattened list of size 1x24, with an approximated q-value for each action. Figure 5a shows a diagram outlining the basics of this architecture.

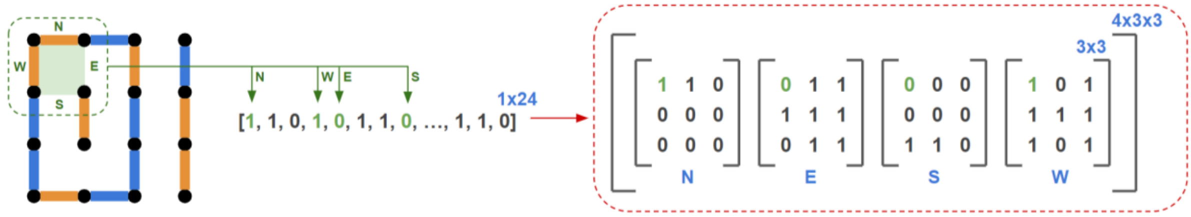


Figure 4: Example of how this board state would be encoded as a 4x3x3 boolean array. The first box, coloured green in this figure, encodes all four of its edges as boolean values, 1 for a placed edge and 0 for an unplaced edge. The boolean edges of each box are encoded in 3x3 matching box placement on board states – with one 3x3 matrix for each NESW coordinate.

CNNs typically lend themselves to image-based supervised learning as each layer applies filters to subsets of the input, sweeping over the input data to create a feature map summarising the whole image. This makes CNNs ideal for any input that contains features with strong local connectivity, but also means the way in which board state data is fed into the network is very important to highlight said local connectivity. For these reasons, Dots and Boxes board states for 4x4 dot grids were reshaped from 1x24 size lists of Boolean values denoting all 24 board edges and their states, where 1 denotes a placed edge and 0 denotes an unplaced edge, into 4x3x3 size boolean arrays based on edge NESW coordinates in relation to each of the 9 boxes (Fig. 4).

3.2.2 Training

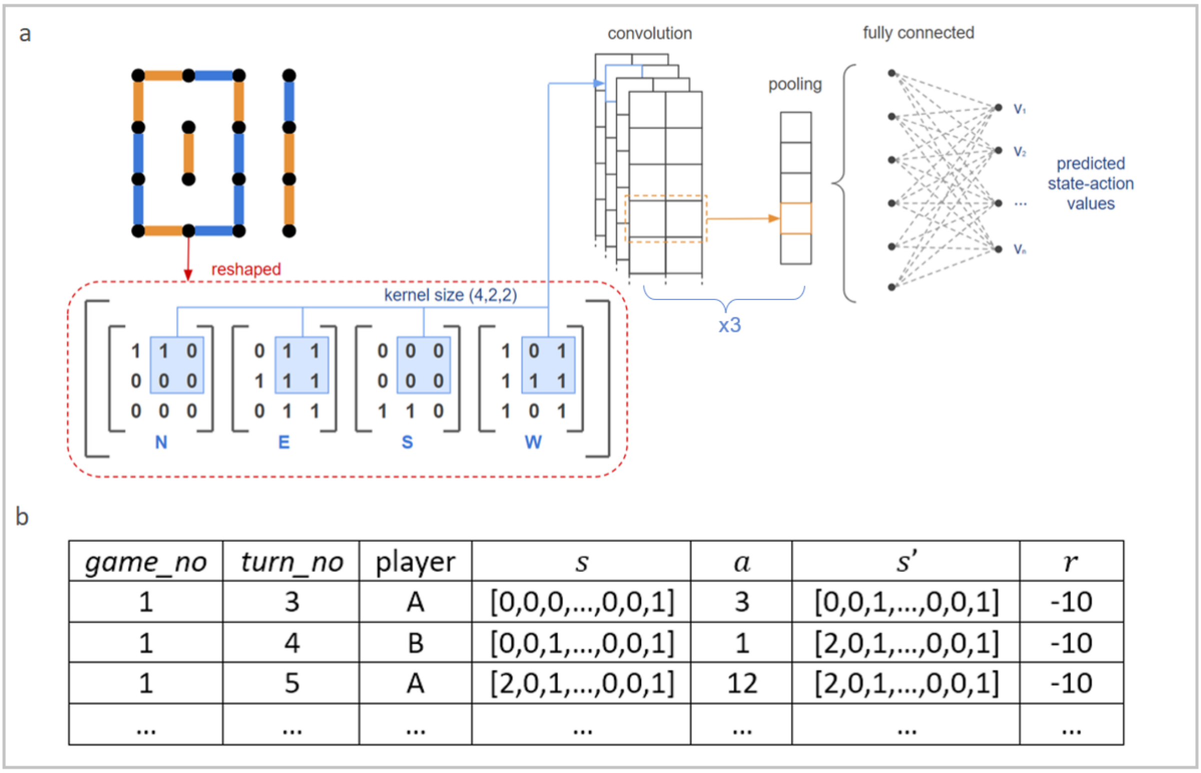
Deep Q-learning neural networks must be trained to approximate the q-value function using large batches of state-action-reward data, often referred to as experience replay data. The training of a CNN consists of a forward phase, where input is passed through the network completely, and a backward phase in which hidden layer weights are updated through backpropagation, where the gradient of the loss function between target and predicted values is calculated with the aim of minimising loss through changing network weights.

In deep q-learning a neural network is used to approximate the Q-value function, so that for every state fed into the neural network a predicted Q-value for each action taken from that state can be made. Instead of directly updating Q-values, as in classical q-learning, neural network weights must be updated at each Bellman update step, similar to q-learning with linear approximation of features. Weights are updated through the loss function, which is the mean squared error of predicted Q-values and target Q-values:

$$L(w) = \left(\left(r + \gamma max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; w^{target})\right) - Q\left(s, a; w^{predicted}\right)\right)^2$$

As the true target Q-values are unknown, with this being a reinforcement learning problem, the regression to the target Q-values can be unstable as the target changes through training. In order to stabilise training, a target Q network is needed separate from the prediction Q network, which changes with each episode. Both networks use the same architecture but the target Q network weights, w^{target}, are updated by the prediction network weights, $w^{predicted}$, after a number of training episodes, allowing the target network to be fixed for periods of time.

Another necessity to stabilise deep-q training is training through experience replay. An experience replay table is required to store all past states and actions. In the early stages of training the deep-q agent plays by its current policy, be that random or a policy trained from a previous iteration. As the agent plays all information on each turn is stored in a table, including total reward updated for each turn at the end if the agent wins or loses (Fig. 5b). Once the table is of a certain size, the agent may then start training from the table, meaning the agent is training via sampling from a uniformly distributed batch of data instead of overfitting on a small number of turns from a single game.



game_no	turn_no	player	s	a	s'	r
1	3	A	[0,0,0,…,0,0,1]	3	[0,0,1,…,0,0,1]	-10
1	4	B	[0,0,1,…,0,0,1]	1	[2,0,1,…,0,0,1]	-10
1	5	A	[2,0,1,…,0,0,1]	12	[2,0,1,…,0,0,1]	-10
…	…	…	…	…	…	…

*Figure 5: **a** CNN architecture; board representations are reshaped into numeric matrices and passed as input through convolution and pooling hidden layers. The resulting predicted q-values for state-action pairs are outputted as a vector. **b** Example experience replay table entries. For each simulated game at each turn, states, actions and rewards are stored until enough data is available to train the CNN in batches.*

To avoid the deep-q agent converging to a locally-optimum policy early exploration was necessary in each training iteration. Epsilon decay was utilised, where epsilon is set as a high value in the initial stages of training and slowly decreases with each episode of training. Each time the agent is prompted to take an action it first checks if a randomly generated number n $(0 \leq n \leq 1)$ is greater than epsilon ε $(0 \leq \varepsilon \leq 1)$. If $\varepsilon > n$ then the agent selects a random action from the available action space, otherwise the agent selects an action based on $argmax(Q)$. At the start of training, the probability of the agent choosing a random action and exploring the state-action space is high (e.g. $\varepsilon = p_{init} = 0.7$).

As training progresses, ε decays at rate r over N training episodes:

$$r = max\left(\frac{N - n_{step}}{N}, 0\right)$$

$$\varepsilon \leftarrow (p_{init} - p_{end})r + p_{end}$$

As training progresses, the agent can then begin focusing on exploitation and converge to a globally-optimum policy rather than a locally-optimum one.

To select the optimal hyper-parameters, the Deep-Q model was run multiple times against a random policy agent over 5,000 games with incremental changes to either the learning rate, α, or discount factor, γ. When training the Deep-Q model against a random policy agent over 5,000 games, $\gamma \geq 0.75$ and initial learning rate $0.75 \leq \alpha_{init} \leq 0.85$ were found to be the optimal boundaries for initial hyper-parameters. $\alpha_{init} \geq 0.9$ displayed a dramatic decline in win rate, possibly due to unstable learning as old learning is continuously rewritten. Ultimately, the hyper-parameters were set as $\alpha_{init} = 0.85$ and $\gamma = 0.9$, the same values as in the q-learning with linear combination of features agent with the same learning rate decay as defined in Section 3.1.2 of this paper.

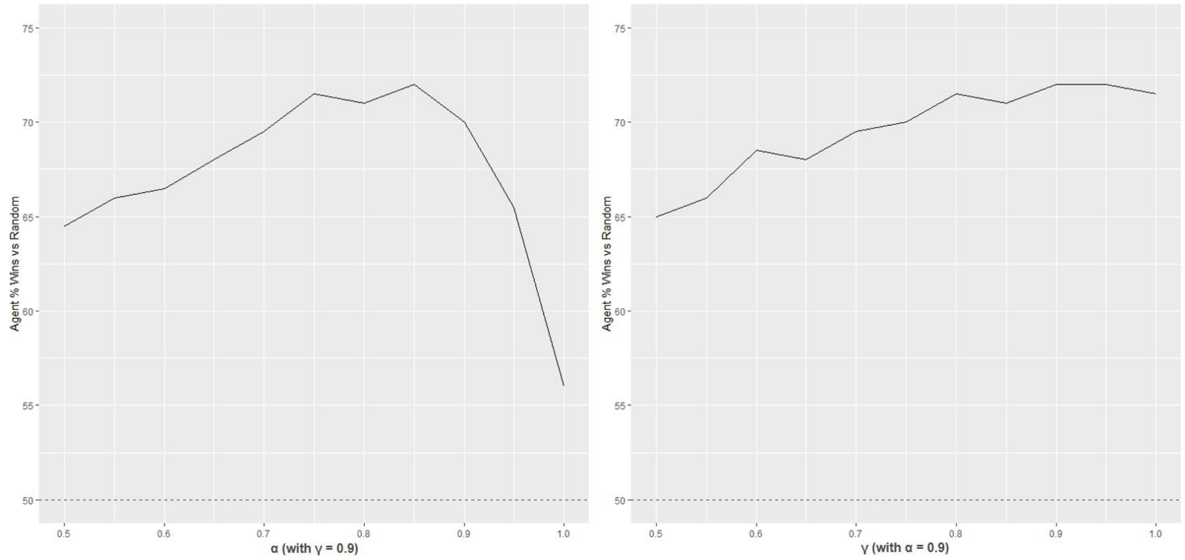


Figure 6: Line plots showing hyper-parameter tests for learning rate (α) and discount factor (γ), where Deep-Q agent % wins against random agent are taken (averaged over 200 games) after training against 5,000 games with the specified parameters. 50% is marked as the baseline % wins that Deep-Q agents start at.

As Dots and Boxes is a 2-player game, an opponent is necessary in the model training process to generate experience replay data. To allow for comparisons to be made between human learners and

Deep-Q agents, both human volunteers and Deep-Q agents learned how to play Dots and Boxes against the q-learning with linear combination of features agent.

There are many different approaches that can be taken when training a neural network, each influencing the resulting agent policy in various ways. As such, multiple Deep-Q models were trained with their resulting agent policies compared to each other. One instance of the Deep-Q model, referred to as Deep-Q agent 1, was trained against the 'box-greedy' q-learning with linear combination of features agent; just as human volunteers learn how to evolve their Dots and Boxes strategy against the 'box-greedy' q-learning agent. A second instance of the Deep-Q model, referred to as Deep-Q agent 2, was trained using a three-step shared learning approach. Shared learning is a training technique where a single network is trained via the use of many adversaries with diverse policies (Zhou et al. 2019). This strategy typically speeds-up training and allows a single agent to develop generalised counter-strategy against a wider range of opponents.

The training of Deep-Q agent 2 was performed in three main stages:

1. A new Deep-Q agent with no prior policy trained against a random policy agent, creating Deep-Q agent 2A.
2. Newly updated Deep-Q agent 2A trained against the q-learning with linear approximation of features agent, creating Deep-Q agent 2B.
3. Newly updated Deep-Q agent 2B trained against a fixed copy of itself (Deep-Q Agent 2B') in competitive multi-agent style, creating Deep-Q agent 2C.

Multi-agent reinforcement learning, where multiple agents cohabit an environment either cooperatively or competitively, is often used to train models but this method can be very time-consuming when both agents begin their training with no prior policy. With the addition of steps 1 and 2 before competitive multi-agent training in step 3, pre-existing agents can be used to accelerate training before the final multi-agent training step where strategy is refined. Training two or more agents adversarially is an oft-used training method in reinforcement learning, where simulating training data is sometimes difficult and creating new adversarial agents to train against is time consuming. TD-Gammon was one of the early applications of multi-agent reinforcement learning and self-training to board games (Tesauro, 1995). Following TD-Gammon, more sophisticated multi-agent methods have been developed to train multiple agents cooperatively and competitively, however for the purposes of this project training Deep-Q agent 2B against a fixed Deep-Q Agent 2B' was the most effective and stable approach.

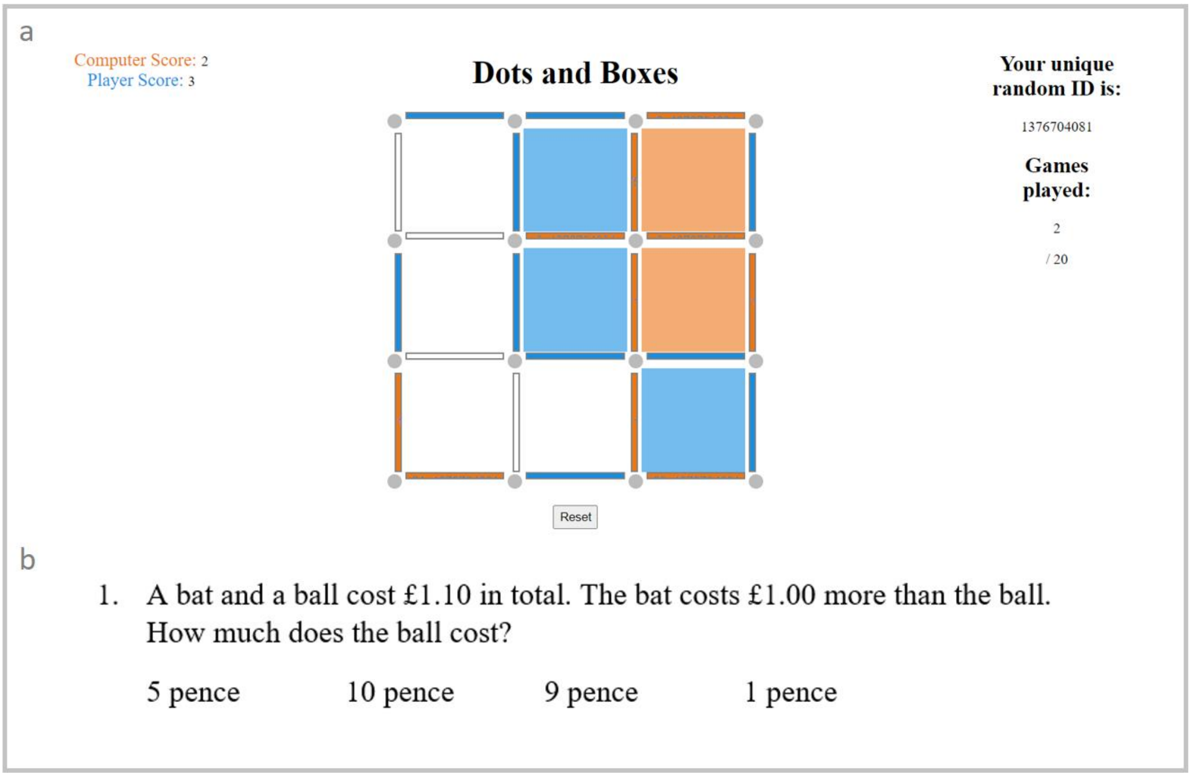
Similar to how prior knowledge and experience aids human learning, machine learning algorithms can benefit from techniques such as transfer learning and training against various opponents. The introduction of prior knowledge and learning in reinforcement learning can be done in many ways depending on the task at hand. Many transfer learning techniques involve first training an agent to complete a similar problem then using that same model as a starting point to train to complete the intended problem. While this technique is more frequently used in image-based classifications and predictions, some studies have displayed benefits of transfer learning from different problems or domains in strategy problems such as games. Sato, Iida and van den Herik have displayed transfer learning by first training an agent in a Tic-Tac-Toe environment then using that prior knowledge to aid the same agent when trained in a Connect 4 then Connect 5 environment (2015). For the purposes of this study, the Dots and Box task is simple enough to use transfer learning against agents of varying levels of strategy without need for a simpler domain or task, such as Tic-Tac-Toe, to draw knowledge from.

3.3 Volunteer Trials

To assess the similarities and differences in human and machine learning, it was necessary to collect human volunteer data of participants playing games of Dots and Boxes, with the intention of comparing the learning process and resulting strategy of individuals across 20 games to the training of q-learning algorithms. In order to more accurately draw comparisons between algorithms and participants, human volunteers played all 20 games against the ‘box-greedy’ q-learning with linear combination of features agent, just as Deep-Q agent 1 was trained.

To collect human volunteer data, a website was created using Python with the Flask web application framework to host the greedy q-learning algorithm and allow volunteers to play against it (Fig. 7a). 71 participants were recruited and forwarded to a website explaining the rules of the Dots and Boxes game. After completing a short 2x1 dot board task to assess understanding of the task, participants were then forwarded to a representation of a 4x4 dot board where they took turns playing against the ‘box greedy’ q-learning with linear combination of features algorithm to place edges and claim boxes across 20 games. Game information including board state, action played, turn number and number of boxes claimed was stored in a data frame at each turn to get a complete log of each volunteers’ games.

The majority of participants were recruited from the Durham University Psychology Department participant pool, from which all participants fall in the age range of 18-24. All participant demographics are reported in Supplementary Table 1. No exclusion criteria were applied in the volunteer process. All volunteers were asked to sign a consent form and were provided with an information sheet detailing the experiment and intended use of data (Supplementary Fig. A & B).



Computer Score: 2
Player Score: 3

Your unique random ID is:
1376704081

Games played:
2
/ 20

1. A bat and a ball cost £1.10 in total. The bat costs £1.00 more than the ball. How much does the ball cost?

5 pence 10 pence 9 pence 1 pence

*Figure 7: **a** The Dots and Boxes task user interface allowing volunteers to complete 20 games against the q-learning agent. **b** An example question from the CRT MCQ 4 test, where four answers are presented to volunteers. 5 pence is the correct answer (measuring system 2), 10 pence is the intuitive answer (measuring system 1) and the other two answers are incorrect decoys.*

Each game played by volunteers is tagged with the level of strategy reached in that game, from levels 1 to 5 based on the phases of strategy described by Weaver and Bossomaier (1998). Each volunteer starts at level 1 (random) and is marked as having progressed to a higher level of strategy if said player exhibits that level of strategy consistently for at least three games in a row. Levels of strategy are decided by the following criteria:

- Level 2 (Box Greedy): Player exhibits this strategy if, as soon as opponent places third edge on a box, the player completes the box.
- Level 3 (Box Greedy +): Player exhibits this strategy if, alongside exhibiting level 2 strategy, they also avoid placing the third edge on a box if possible.
- Level 4 (Temporal): Player exhibits this strategy if, alongside exhibiting levels 2 and 3 strategy, they also bait the opponent into claiming short chains instead of long chains.
- Level 5 (Temporal +): Player exhibits this strategy if, alongside exhibiting levels 2, 3 and 4 strategy, they also sacrifice the last two boxes in a chain to bait the opponent into completing the two boxes and opening up a longer chain.

When studying human decision-making and learning, it is important to avoid making assumptions that all individuals are alike and behave as a group. A multitude of factors can influence the way individual humans make decisions and learn, from personality attributes such as drive and decisiveness to general intelligence. For these reasons, it was important to try to assess the sensitivity to reward and punishment in human volunteers as well as individual use of systems 1 and 2 thinking. After playing 20 games, volunteers were asked to complete the BIS/BAS questionnaire and the CRT-MCQ 4 test to measure sensitivity to punishment and reward and use of system 1 and 2 thinking respectively. Both Dots and Boxes task data and questionnaire data were stored with unique random IDs to map task data to questionnaire responses while still maintaining volunteer anonymity. Once collected, answers to the BIS/BAS questionnaire and CRT-MCQ 4 test were graded and scores for individuals were calculated according to the guidance from their original papers.

Results

4.1 Volunteer Trial Data

In total 71 responses were received, 9 of which were excluded due to incompletion of either the Dots and Boxes task or questionnaires, leaving 62 participants with complete datasets to perform analysis on. Participants were classed as high scorers for BIS/BAS groups if satisfying $\bar{x} \geq \mu$, where $\bar{x}$ is individual mean score of BIS/BAS group questions and μ is mean of population. Participants were classed as high scorers for CRT groups if satisfying $s \geq \mu$, where s is score of CRT groups and μ is the population mean score. Data across BIS/BAS and CRT was fairly bimodal and as such it made sense to place participants in categories based on this. Distributions of participant scores can be found in Supplementary Figures 3a-f.

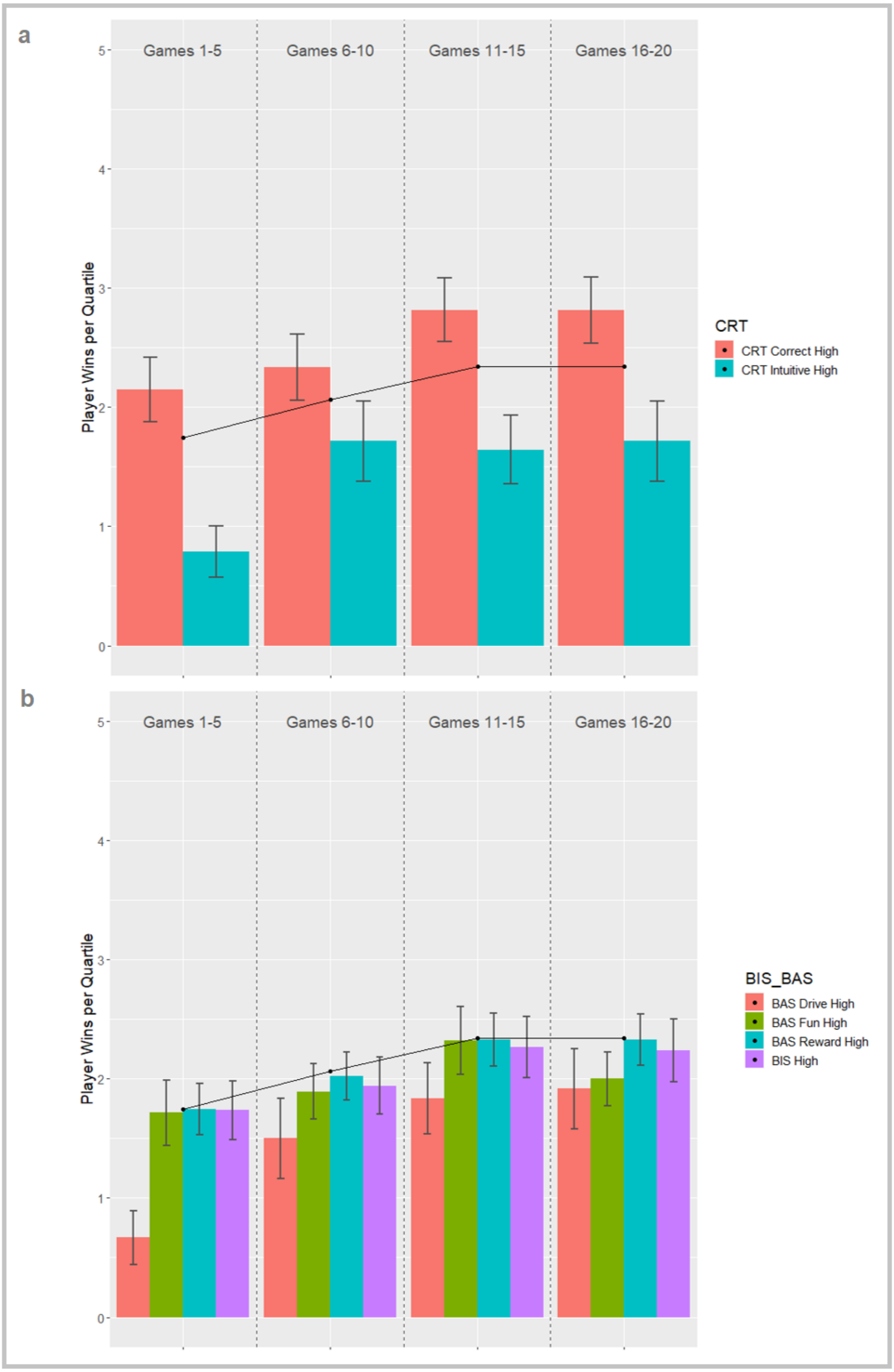


Figure 8: ***a & b*** *Line plot displays average number of player wins per quartile (5 games per quartile) across all 62 participants. Bar plots display average number of player wins per quartile for each BIS/BAS and CRT group. Standard error displayed for each bar. CRT Correct High n = 38, CRT Intuitive High n = 32, BIS High n = 32, BAS Drive High n = 32, BAS Reward High n = 34, BAS Fun High n = 27.*

As observed in Figure 8a, participants classed as high correct CRT scorers won more games in each quartile than average, and participants classed as high intuitive CRT scorers won fewer games in each quartile than average. Interestingly, while number of games won per quartile plateaus in quartiles 3 and 4 across all 62 participants, there is a decline in number of games won from quartile 3 to quartile 4 among participants with high BAS Fun Score. While this could seem to suggest a loss of interest and subsequent carelessness in gameplay by the fourth quartile of games among high BAS Fun scorers, when running additional two-way ANOVA tests, the impact of high BAS Fun on total score across all quartiles was not statistically significant, as was the TukeyHSD pairwise tests between quartiles 3 and 4 in high BAS Fun participants (Supplementary Tables 3 & 4). Only high CRT correct, CRT intuitive and BAS Drive displayed a statistically significant effect on participant total won games (Supplementary Table 3).

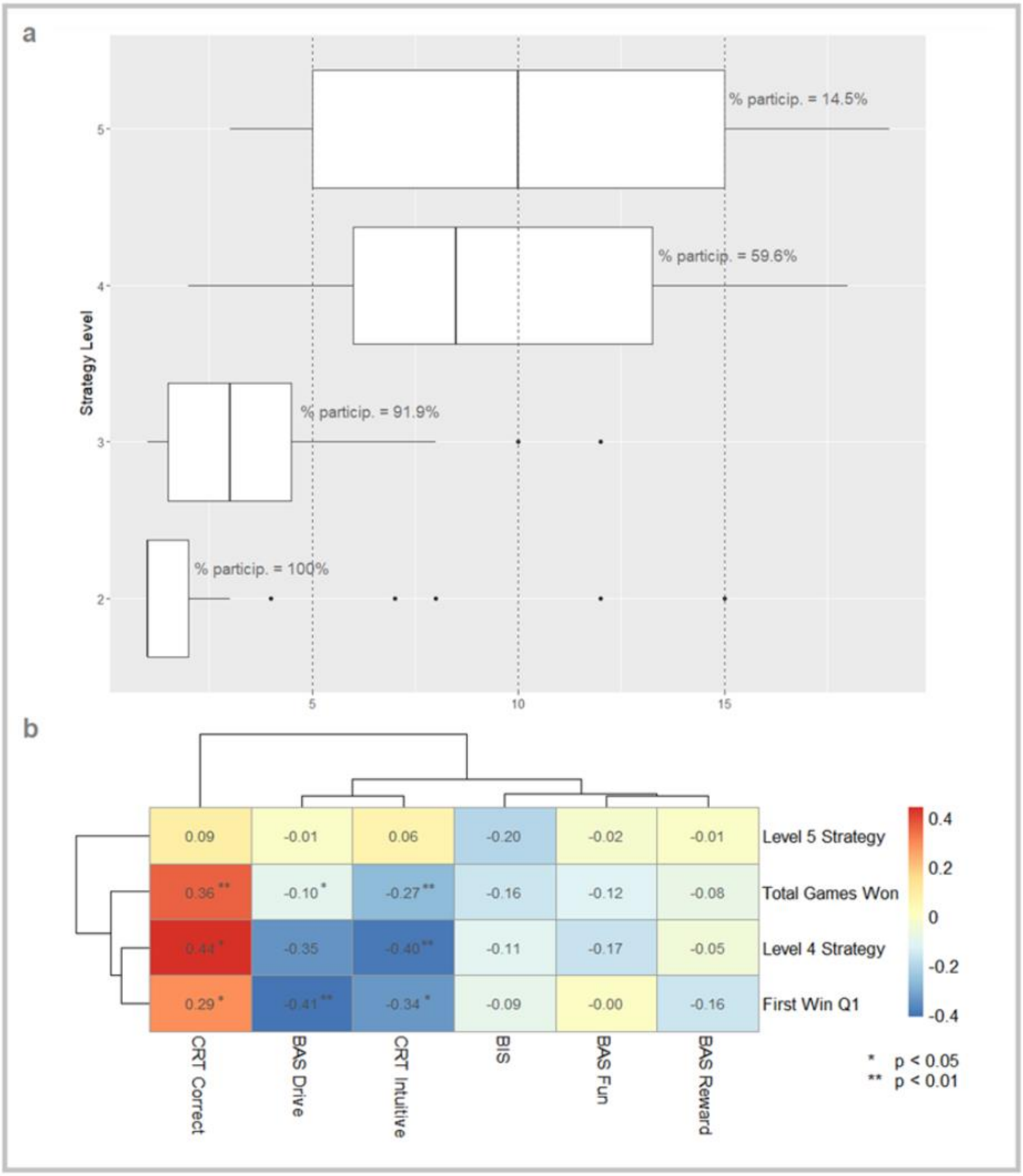


*Figure 9: **a** Box plots displaying distribution of games at which players first reach each level of strategy between levels 1-5, where level 1 is the level all participants start at. The percentage of the participants exhibiting each level of strategy is displayed to the right of each box plot. **b** Heatmap displaying Spearman correlation between BIS/BAS and CRT groups and game metrics. Rho (r values) displayed between each group with significance of each correlation denoted by asterisks. Strategy Level 3 n = 57, Level 4 n = 37, Level 5 n = 9.*

Each game played by participants was tagged with a level of strategy reached as outlined in the Methods section of this paper. All participants were able to reach level 2 of strategy before completing all 20 games, and the majority of participants (91.5%) were able to reach level 3 of strategy before completing all 20 games. Fewer participants reached levels 3 and 4 of strategy before completing all 20 games (59.6% and 14.9% respectively). The majority of participants reached levels 2 and 3 strategy within the first quartile (games 1 to 5) of games

As observed in Figure 9b, there is a moderate positive correlation of rho(36)=0.36, p<0.01 between participants classed as high correct CRT scorers and total number of games won. Additionally, there is a moderate positive correlation of rho(36)=0.44, p<0.05 between high correct CRT scorers and number of games played at level 4 strategy, and low positive correlation of rho(36)=0.29, p<0.05 between high correct CRT scorers and players winning their first game between games 1 to 5. There is a moderate negative correlation of rho(30)=-0.40, p<0.01 between participants classed as high intuitive CRT scorers and number of games played at level 4 strategy. Additionally, there is a moderate negative correlation of rho(30)=-0.34, p<0.05 between high intuitive CRT scorers and players winning their first game between games 1 to 5, and a low negative correlation of rho(30)=-0.27, p<0.01 between high intuitive CRT scorers and total number of games won.

There are no correlations between number of games played at level 5 strategy and BIS/BAS or CRT groups, bar the low negative correlation between high BIS scorers and employment of level 5 strategy, however this is not a statistically significant correlation. There is a moderate negative correlation of rho(30)=-0.41, p<0.01 between high BAS Drive scorers and players winning their first game between games 1 to 5, and low negative correlation of rho(30)=-0.10, p<0.05 between high BAS Drive scorers and total number of games won. There is also a moderate negative correlation between high BAS Drive scorers and number of games played at level 4 strategy, however the correlation is not statistically significant. Interestingly, there is no significant correlation between groups high BIS, high BAS Fun and high BAS Reward against game and strategy metrics.

4.2 Q-Learning with Linear Combination of Features

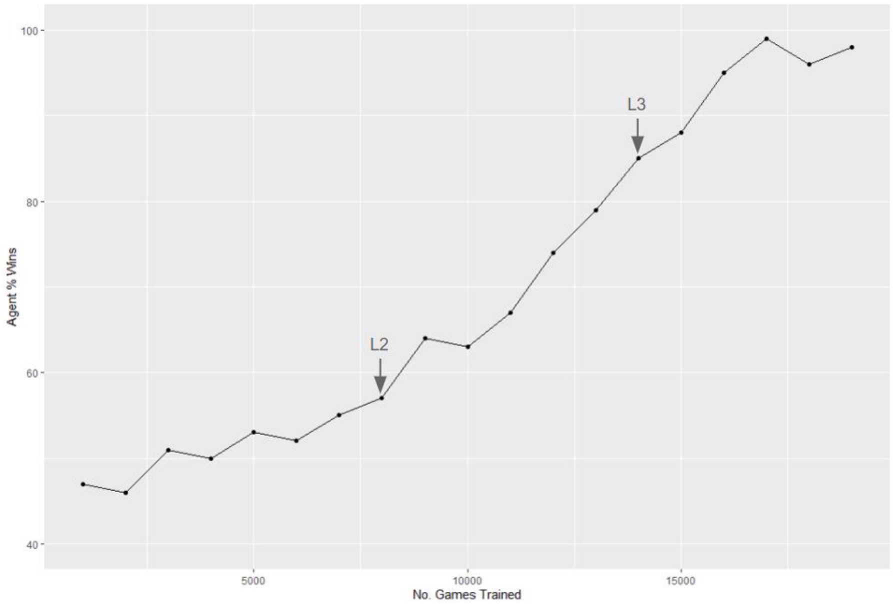


Figure 10: Line plot displaying percentage games won after each 1,000 games trained against, where training is between a Q-learning with linear combination of features agent against a random agent. Levels of strategy reached at training stages are marked.

The first agent trained was the q-learning with linear combination of features agent. A random policy agent was created to play against the q-learning model, with the q-values of the approximated feature-action space updated at the end of each game depending on overall result of the game, with all feature-action pairs either rewarded or penalised for winning or losing respectively, as well as a smaller additional reward for individual feature-action pairs that complete a box.

Training was performed until the agent had converged to a globally optimum policy and win rate had stabilised over 19,000 games against the random policy agent. At the end of each interval of 1,000 games, 100 games were played between the random agent and q-learning agent independently of training to track progress and collect metrics such as percentage wins.

A random sample of 20 games from each 100-game test phase was taken and strategy for each game was logged against rules defined in Section 4.1. If the agent displayed a level of strategy as defined consistently across all 20 games, then that phase of training is marked as having reached that level. The q-learning agent consistently displayed stage 2 strategy after training against 8,000 games, then stage 3 after 14,000 games, at which point win rate had stabilised, as seen in Figure 10. Stages 4 and 5 were not reached, leading to the q-learning agent settling on a ‘box greedy’ policy.

4.3 Deep-Q Learning

Two versions of Deep-Q learning agents with identical model architecture were trained to both investigate differences in training approaches and ultimately select the best Deep-Q agent between the two. Deep-Q agent 1 was trained solely against the q-learning with linear combination of features agent, and Deep-Q agent 2 was trained in three stages against a random policy agent (creating Deep-Q agent 2A), then against the q-learning with linear combination of features agent (creating Deep-Q agent 2B), then finally against a fixed copy of itself (creating Deep-Q agent 2C). Deep-Q agent 1 was trained until performance plateaued at around an 84% win rate when against the q-learning with linear combination of features agent. Deep-Q agent 2 was trained in three phases, where training moved on to the next phase once performance began to plateau, ultimately reaching a win rate of around 84% in phase 3 of training, when training against a fixed copy of itself.

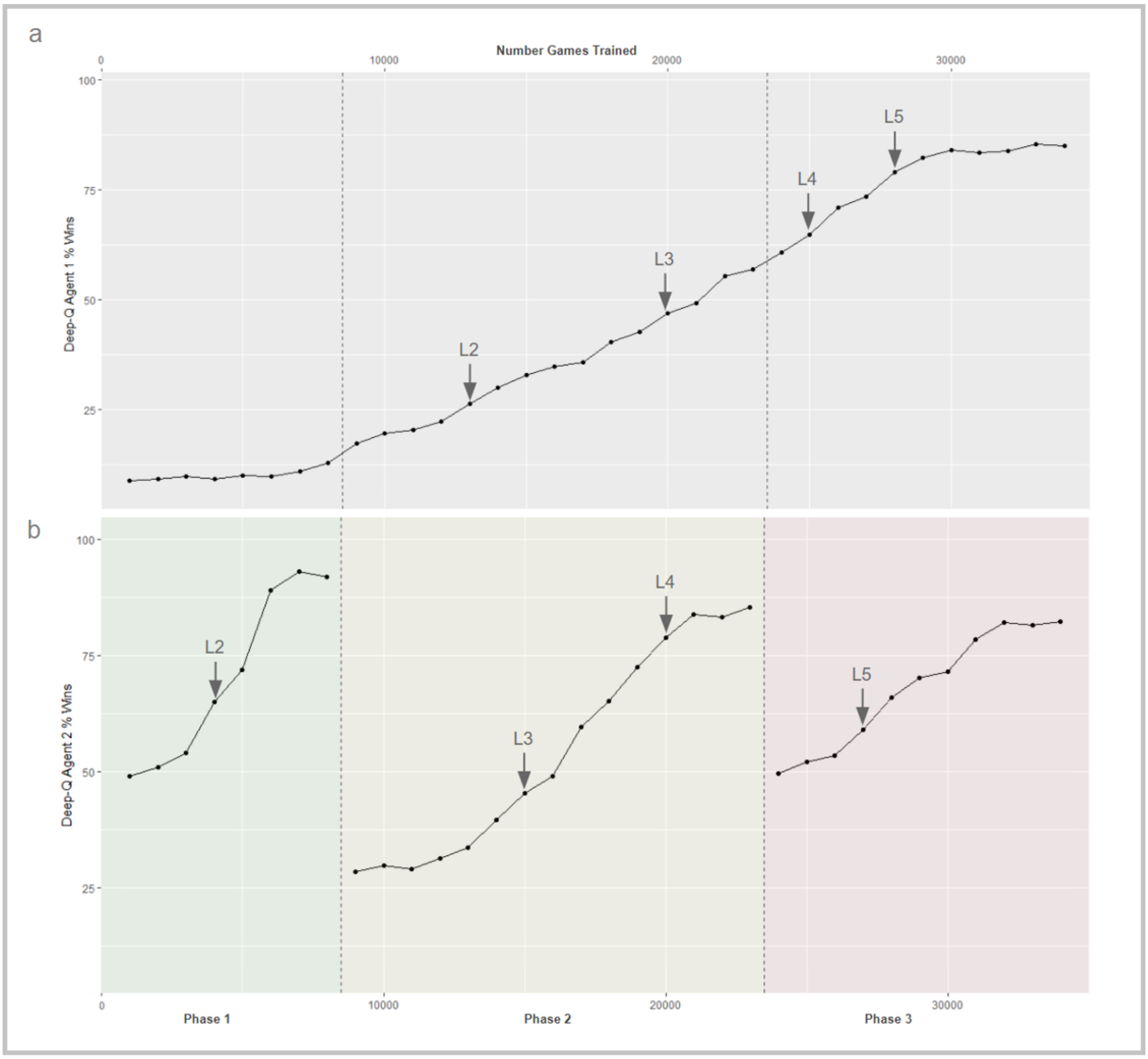


Figure 11: ***a*** *Line plot displaying percentage games won after each 1,000 games trained against, where training is between a Deep-Q learning agent against a q-learning with linear combination of features agent.* ***b*** *Line plot displaying percentage games won after each 1,000 games trained against, where training is done in three successive phases on the same model. Phase 1 training is performed between a Deep-Q learning agent against a random agent, phase 2 against a linear combination of features agent and phase 3 against a copy of the Deep-Q learning agent. Levels of strategy reached at training stages are marked.*

Similar to the analysis of strategy in Section 4.2, a random sample of 20 games from each 100-game test phase was taken and strategy for each game was logged against rules defined in Section 4.1. If the agent displayed a level of strategy as defined consistently across all 20 games, then that phase of training is marked as having reached that level. Deep-Q agent 1 consistently displayed stage 2 strategy after training against 13,000 games, stage 3 strategy after 20,000 games, stage 4 after 25,000 games and stage 5 after 28,000 games. Deep-Q agent 2 consistently displayed stage 2 strategy after training against 4,000 games, stage 3 strategy after 15,000 games, stage 4 strategy after 20,000 games and stage 5 strategy 27,000 games. Deep-Q agent 2 displayed all levels of strategy earlier in training than Deep-Q agent 1, suggesting training in transfer learning phases allowed Deep-Q agent 2 to train faster. Ultimately, both Deep-Q agents consistently displayed a ‘temporal’ policy towards the end of training.

Ultimately, Deep-Q agent 2 was selected as the final Deep-Q agent as it not only displayed temporal strategy stages 4 and 5, but also was able to display lower levels of strategy when necessary, for example when playing against a random policy agent. While Deep-Q agent 1 trained only against the q-learning with linear combination of features agent was able to display stages 4 and 5 of strategy, this display of higher-level strategy was often less consistent and the agent struggled to generalise against unfamiliar opponents, such as a random policy agent.

4.4 Volunteer Learning and Deep-Q Learning

Drawing comparisons between the learning process of human volunteers and the training process of RL agents is fraught with difficulties. Volunteers had only 20 games to develop their strategy from a random policy, whereas RL agents had thousands of simulated games to draw experience from. However, this comparison is also clouded by the fact that humans have decision-making experience outside of the artificial Dots and Boxes Task environment, aiding their learning of the task with transferable skills and quick understanding of the game environment - RL agents often do not have such experience to draw from.

The resulting three agents created from q-learning can be described as:

- Random (with no policy learned via q-learning), where the agent places edges randomly for all board states.
- Box Greedy (with policy learned from q-learning with linear combination of features), where the agent typically avoids placing the third edge on a box if it can and will always place the last edge on a box if the opportunity presents itself.
- Temporal (with policy learned from Deep Q-learning), where the agent utilises a similar strategy to ‘box greedy’ but will not always try to claim boxes if sacrificing a small number of boxes can lead to claiming more boxes in the future.

With these three distinct strategies, comparisons can be made between machine learning agents and human volunteers with their own distinct strategies. To draw said comparisons, volunteer game play must be benchmarked against each agent. For each volunteer game, the board states of each player turn are reshaped into matrices and passed through each of the three agents to find the resulting predicted q-values from each of the random, greedy and temporal policies. Comparisons can then be made between actions made by human volunteers and actions that would have been made by each of the three machine learning agents.

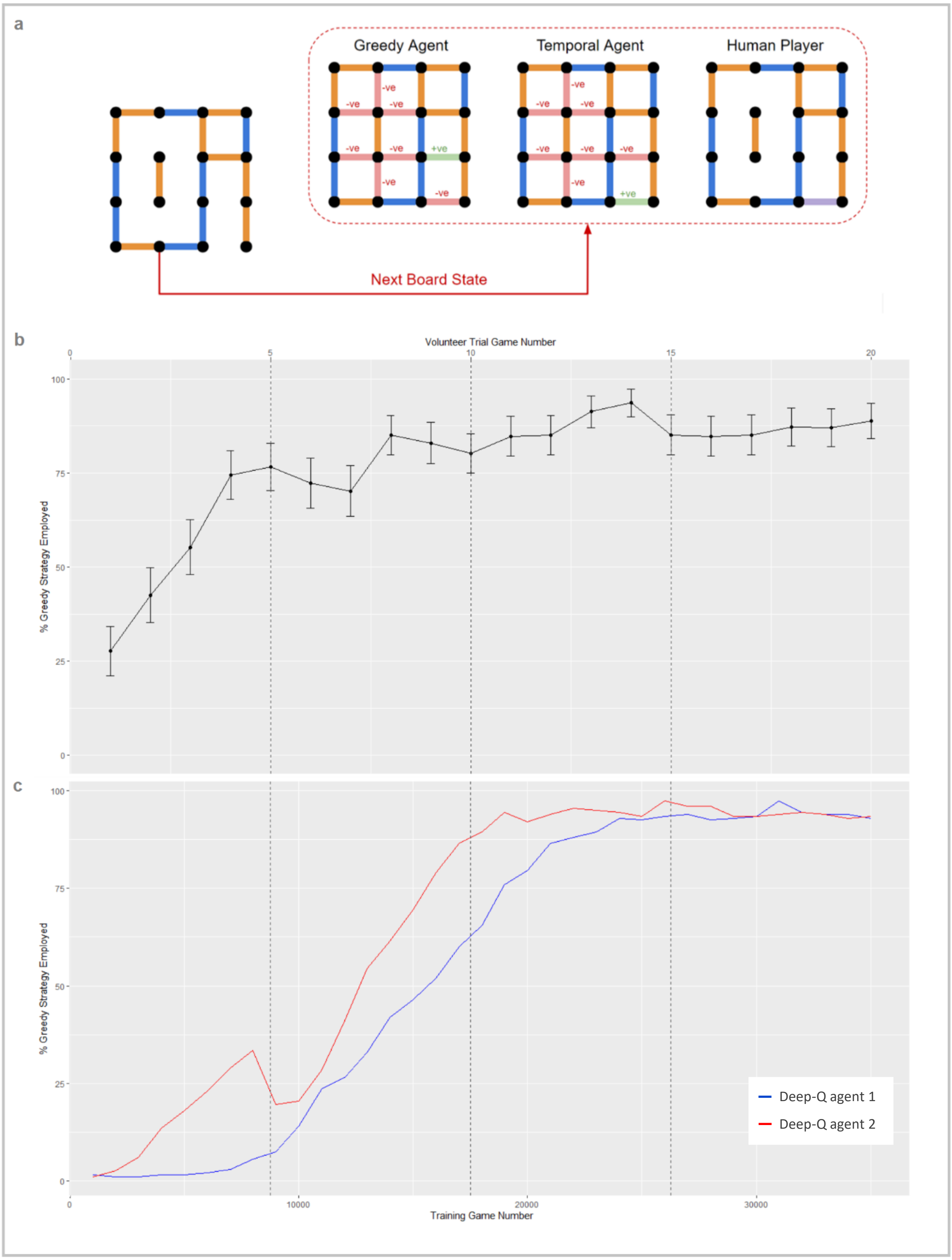


*Figure 12: **a** Example board state showing predicted q-values of each agent strategy and actual strategy of human player. In this example, the human player seems to adopt a temporal strategy as opposed to a greedy one.**b** Line graph showing the percentage of participants using greedy (or higher level) strategy in each game over the 20 trial games, based on correlation between volunteer gameplay and greedy agent predicted q-values. Standard error bars displayed. **c** Line graph showing the percentage of test games using greedy (or higher level) strategy at each 1,000 training game mark of both Deep-Q agent 1 and Deep-Q agent 2, based on correlation between deep agent gameplay and greedy agent predicted q-values.*

Figures 12b and 12c show the learning and adoption of greedy or higher strategy across human volunteer trials and deep learning agent trainings. Volunteer board states across all 20 games were fed into the resulting greedy and temporal agents, where q-values for each state-action pair were predicted and compared to the state-action pairs individual volunteers actually played. Random, greedy or temporal strategy was assigned to each game from each individual volunteer based on the highest correlating agent strategy across all board states. This same process was applied to Deep-Q agents, where 200 test games were played at each 1000 games training mark, q-values for each state-action pair were predicted using previously trained agents and strategy was assigned based on which agent correlated with training decisions.

Human volunteers showed rapid learning of greedy strategy within the first quartile of games played, after which point the adoption of greedy strategy or higher seemed to fluctuate around 70% of games played. In contrast, both Deep-Q agents displayed more gradual learning and adoption of greedy strategy, Deep-Q agent 1 more so than Deep-Q agent 2. Additionally, the use of greedy or higher strategy was more consistent in both Deep-Q agents once greedy strategy was learnt when compared to volunteer games. There is a decline in adoption of greedy strategy at the 9,000 training games mark for Deep-Q agent 2, however this is likely due to the agent switching from training against a random agent to training against the q-learning with linear combination of features agent at that point.

Between the two Deep-Q agents, while there are still stark differences, Deep-Q agent 2B (the resulting Deep-Q agent when trained against the q-learning with linear combination of features agent) most closely resembles the human volunteer learning process. One of the many difficulties in drawing comparisons between human and machine learning is the lack of context and prior learning in machine learning agents. Human volunteers likely will have played games similar to dots and boxes before, and as such will be able to transfer prior learning to the Dots and Boxes environment; depending on how agents are trained, there is little to no context and a policy must be learned from nothing. As Deep-Q agent 2B was first trained against a random Dots and Boxes agent before being trained against the greedy q-learning with linear approximation of features agent, it has more prior information then Deep-Q agent 1, and as such is able to win more early-stage games as well as learning and adopting greedy strategy rapidly.

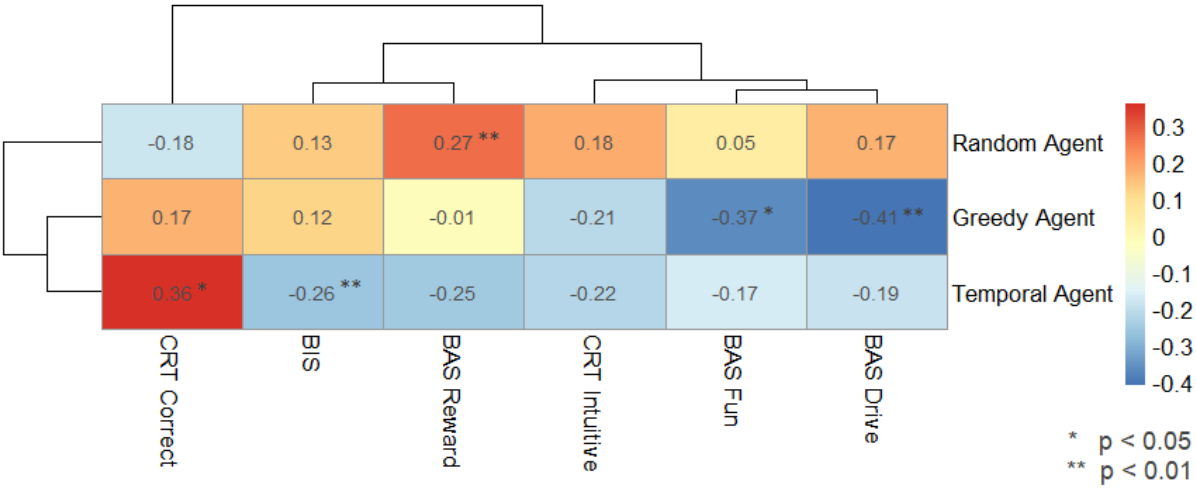


Figure 13: Heatmap displaying Spearman correlation between BIS/BAS and CRT groups and predicted q-value agent strategy groups. Rho (r values) displayed between each group with significance of each correlation denoted by asterisks.

As observed in Figure 13, there is a moderate positive correlation of rho(36)=0.36, $p<0.05$ between high correct CRT scorers and temporal agent strategy. While there are low negative correlations between high intuitive CRT scorers and both temporal and greedy agent strategies, these correlations are not statistically significant. There is a low negative correlation of rho(30)=-0.26, $p<0.01$ between high BIS scorers and temporal agent strategy. Additionally, there is a low positive correlation of rho(32)=0.27, $p<0.01$ between high BAS Reward scorers and random agent strategy. Both high BAS Fun and high BAS Drive scorers have moderate negative correlations, rho(25)=-0.37, $p<0.05$ and rho(30)=-0.41, $p<0.01$ respectively, with greedy agent strategy.

Discussion

The Dots and Boxes game proves to be an effective task to study human and machine learning, allowing for observation of the impact of behavioural factors such as inhibition, sensitivity to reward and systems 1 and 2 thinking on learning and decision making. Additionally, through the study of learning progression in both human participants and machine learning agents, the clear benefits of transfer learning in humans can be observed.

While it is seen in many machine learning studies that transfer-learning or prior knowledge in models indeed speeds up training when applied effectively, it is interesting to confirm this in this study while also drawing comparisons between prior knowledge in reinforcement learning agents and in human volunteers when learning against the same opponent. Despite Deep-Q agent 1 learning how to play Dots and Boxes against the same q-learning with linear combination of features agent as human volunteers, Deep-Q agent 2's learning progression observed in Figure 9 bears much more similarity to human volunteer learning progression observed in Figure 7c, where level 2 strategy is learned early on and adoption of levels 3-5 strategy gradually occurs as training progresses. While Deep-Q agent 1 plateaus in win rate before Deep-Q agent 2, as seen in Figure 11, it is important to remember Deep-Q agent 2 is at this point training against a more advanced opponent than Deep-Q agent 1; as such, reaching milestones such as level of strategy is a better indicator of training speed. Deep-Q agent 2 displays consistent use of level 5 strategy earlier than Deep-Q agent 1, as seen in Figure 11, suggesting faster training with the aid of transfer-learning. In future studies, it would be interesting to see if transfer-learning using similar board games such as Tic-Tac-Toe before training agents in a Dots and Boxes task environment would improve learning speed further.

Human participants quickly learn and develop a box-greedy strategy, with over 75% of games employing a box-greedy strategy by the 5th game played, observed in Figure 12a. In comparison, both Deep-Q agent 2 and Deep-Q agent 1 displayed much slower adoption of this strategy, with both agents passing the 75% greedy strategy games mark in quartiles 2 and 3 respectively, observed in Figure 12b. However, despite rapid adoption of levels 2 and 3 strategies, only 59.6% of participants consistently displayed level 4 strategy and only 14.9% of participants consistently displayed level 5 strategy, seen in Figure 9a. It is difficult to say if more participants would reach higher levels of strategy if they were studied over more than 20 games or if most participants simply satisficed, deciding that their reached level of strategy provided a 'good enough' win rate and thus not striving to reach higher levels of strategy. In future studies, satisficing may be assessed by increasing the number of games participants must complete to see if level of strategy reached plateaus.

It would be possible to increase model performance and allow agents to develop higher levels of strategy earlier if 2-ply or 3-ply elements were introduced as part of Bellman update steps, where ply is defined as number of moves ahead (Samuel, 1959). Some of the best Deep RL agents apply deeper searching via methods such as Monte Carlo Tree Search (Guo et. al, 2014). The original Bellman equation is only

1-ply, as it only includes maximum reward from the next state and looks no further. This one step of lookahead can limit training performance, especially in tasks where more than one step of lookahead is necessary in higher levels of strategy. Deep Blue would typically search between 6 to 16-ply states in chess game trees to find the optimal strategy (Campbell, Hoane & Hsu, 2002), and TD-Gammon employs 2-ply search to find the optimal Backgammon strategy (Tesauro, 1995). While no more than 2/3-ply is necessary for Dots and Boxes, as it is a simple task compared to the likes of chess, it would still be interesting in future studies to assess agents applying different levels of lookahead compared against human volunteer strategy, especially between CRT system 1 & 2 thinking groups of volunteers. Higher levels or strategy such as levels 4 and 5 require players to look at least two moves ahead, and certain volunteers consistently exhibited higher levels of strategy suggesting their engagement of some degrees of lookahead beyond the next immediate move.

Both the BIS/BAS questionnaire and CRT-MCQ 4 test data allow for interesting analysis into individual learning and decision-making and the behaviours that may drive them. As BIS measures sensitivity to punishment or lack of reward, it is often hypothesised that those with a high BIS score would perform better than average. Indeed, the win rate of high BIS individuals in gambling style games is observed to be higher than the win rate of low BIS individuals (Kim & Lee, 2021). However, it is possible that BIS score becomes unreliable as a predictor of performance in artificial strategy situations such as Dots and Boxes, where poor performance has a much smaller impact on the participant. While BIS score had no observed statistically significant relationship with Dots and Boxes task metrics, such as total games won or games played at level 4 or 5 strategy (see Figure 9b), a low negative correlation was found between high BIS score and the use of temporal agent strategy (see Figure 13). As temporal strategy requires sacrificing short-term gains for long term rewards in the form of giving up short chains of boxes to the opponent, this correlation could potentially indicate high BIS individuals avoid the 'risky' short-term losses despite the possible long-term gains.

Interestingly, both BAS Fun and BAS Drive display moderate negative correlations with greedy agent strategy in Figure 13. When training the 'box-greedy' q-learning with linear combination of features agent, level 3 strategy was reached before the agent satisficed and settled on a strategy that was good enough to achieve a decent win rate with the approximated state space. On the contrary, Deep-Q agent 2's temporal strategy was achieved through maximising where the best strategy was found against a range of opponents and without approximating state space to a simplified representation of the task. In existing studies assessing both BIS/BAS and maximisation vs satisficing the relationship between BAS subtypes such as Drive and Fun are ambiguous. Spunt et al. report positive correlation between BAS Drive and maximising behaviour in participants (2009), however this correlation is low and there is no observed relationship between BAS Fun and maximisation. While the moderate negative correlation between high BAS Drive participants and greedy strategy could be due to high BAS Drive individuals being less likely to satisfice, this hypothesis does not explain the low negative correlation between high BAS Drive participants and temporal strategy, which is maximising in nature. As such, in future studies comparing satisficing in RL agents against human learners the Maximisation Scale (MS) developed by Schwartz et al. may be a more appropriate behaviour metric than assessing BIS/BAS (2002).

While both high BAS Fun and high BAS Drive display moderate negative correlations with greedy agent strategy in Figure 13, high BAS Reward displays no significant relationship with greedy agent strategy and instead shows moderate positive correlation with random agent strategy. Additionally, high BAS Reward displays no significant relationship with game metrics such as total wins or levels of strategy, seen in Figure 9b; while high BAS Reward individuals are sensitive to reward stimuli gained from winning, strategy employed by said individuals is not conducive to the achievement of rewards.

The Cognitive Reflection Test proves to be a useful behavioural metric when assessing individual Dots and Boxes strategy. Strategy of individuals with a higher number of CRT-correct responses shows moderate positive correlation with the strategy employed by the temporal agent, seen in Figure 13, and high CRT-correct individuals also displayed moderate positive correlation with game metrics suggesting higher performance, such as higher total number of wins and number of games played at level 4 strategy, as observed in Figure 9b. These findings consolidate high CRT-correct individual use of system 2 reasoning as opposed to system 1 intuitive thinking to avoid impulsive strategy with low pay-out, instead applying reasoning to adopt a more successful temporal-style strategy. High CRT-intuitive individuals didn't significantly correlate with any of the three RL agent strategies, making conclusions as to strategy employed by these individuals difficult. High CRT-intuitive individuals correlated with game metrics suggesting lower performance, as seen in Figure 9b, and lower performance when compared against the average win rate across participants in each quartile of games can be observed in Figure 8a.

When drawing comparisons between human and machine learning it is important to note that there are orders of magnitude in difference in training times of agents compared to learning times in humans. More can be done to compare agent policies against human volunteer decisions, for example by calculating the Elo ratings of both human and agent players or comparing human volunteer moves against agent policies in similar situations, as displayed in the analysis of AlphaGo against champion Go players (Silver et al., 2018). Another more quantitative approach may have included calculating the likelihood of each human action according to each of the three agents developed and finding the combined likelihood of any given action to better benchmark human learning against machine learning.

Conclusion

To conclude, Dots and Boxes is an effective task to study both human and machine learning, allowing for various machine learning architectures and training methods as well as allowing for observation of how behavioural factors such as systems 1 and 2 of thinking affect human decision making. By benchmarking learning progression in terms of defined levels of strategy reached by both human participants and machine learning agents, the benefits of simulating the transfer learning done by humans in Deep-Q learning agents can be observed. BIS/BAS and CRT-MCQ 4 participant data allow for interesting insights into how behavioural metrics, such as impulsivity vs reasoning and sensitivity to reward and punishment, affect learning and decision making in humans.

References

Alderton, E., Wopat, E. and Koffman, J., 2019. Reinforcement Learning for Connect Four. *Student Report. Stanford University, CA, USA*. https://web.stanford.edu/class/aa228/reports/2019/final106.pdf

Bossomaier, T. and Knittel, A., 2006. An Evolutionary Agent Approach to Dots-And-Boxes. *University of Sydney, Australia. ICSE 2006.*

Campbell, M., Hoane, A. and Hsu, F., 2002. Deep Blue. *Artificial Intelligence,* 134(1-2), pp.57-83. https://doi.org/10.1016/S0004-3702(01)00129-1

Carver, C. S., & White, T. L. (1994). Behavioural Inhibition, Behavioural Activation, and Affective Responses to Impending Reward and Punishment: The BIS/BAS Scales. *Journal of Personality and Social Psychology*, 67, 319-333. https://doi.org/10.1037/0022-3514.67.2.319

Dubey, R., Agrawal, P., Pathak, D., Griffiths, T. and Efros, A., 2018. Investigating Human Priors for Playing Video Games. University of California, CA, USA. *ICML 2018.*

Even-Dar, E. and Mansour, Y., 2003. Learning Rates for Q-learning. *Journal of Machine Learning Research,* 5, pp.1-25. https://doi.org/10.1007/3-540-44581-1_39

Fukushima, K., 1980. Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biological Cybernetics,* 36(4), pp.193-202. https://doi.org/10.1007/BF00344251

Guo, X., Singh, S., Lee, H., Lewis, R. and Wang, X., 2014. Deep Learning for Real-Time Atari Game Play Using Offline Monte-Carlo Tree Search Planning. *University of Michigan, MI, USA. NIPS 2014.*

Hilgard, E., Kimble, G. (Ed.) and Marquis, D., 1961. Conditioning and Learning. London, UK: *Methuen & Co.*

Hubel, D. and Wiesel, T., 1962. Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex. *The Journal of Physiology*, 160(1), pp.106-154. https://doi.org/10.1113/jphysiol.1962.sp006837

Jordan, M. and Mitchell, T., 2015. Machine Learning: Trends, Perspectives, and Prospects. *Science,* 349(6245), pp.255-260. https://doi.org/10.1126/science.aaa8415

Kaelbling, L., Littman, M. and Moore, A., 1996. Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, 4, pp.237-285. https://doi.org/10.1613/jair.301

Kahneman, D., 2003. A Perspective on Judgment and Choice: Mapping Bounded Rationality. *American Psychologist,* 58(9), pp.697-720. https://doi.org/10.1037/0003-066x.58.9.697

Kannengiesser, U. & Gero, J., 2019. Empirical Evidence for Kahneman's System 1 and System 2 Thinking in Design. *University of North Carolina at Charlotte, NC, USA. Human Behaviour in Design.*

Kim, D. and Lee, J., 2021. Effects of the BAS and BIS on Decision-Making in a Gambling Task. *Personality and Individual Differences* 50(7), pp.1131-1135. https://doi.org/10.1016/j.paid.2011.01.041

Kuhl, N., Goutier, M., Baier, L., Wolf, C. and Martin, D., 2020. Human vs. Supervised Machine Learning: Who Learns Patterns Faster? *Karlsruhe Institute of Technology, Germany. ArXiv.* https://doi.org/10.48550/arXiv.2012.03661

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W. and Jackel, L., 1989. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4), pp.541-551. https://doi.org/10.1162/neco.1989.1.4.541

Lee, K. and Mahajan, S., 1990. The Development of a World Class Othello Program. *Artificial Intelligence*, 43(1), pp.21-36. https://doi.org/10.1016/0004-3702(90)90068-B

Melo, F. and Ribeiro, M., 2007. Convergence of Q-Learning with Linear Function Approximation. *Proceedings of the 2007 European Control Conference*, Kos, Greece. pp.2671-2678. https://doi.org/10.23919/ECC.2007.7068926

Mnih, V., Kavukcuoglu, K., Silver, D. *et al.*, 2015. Human-Level Control Through Deep Reinforcement Learning. *Nature* 518, pp.529–533. https://doi.org/10.1038/nature14236

Moeller, F.G., Barratt, E.S., Dougherty, D.M., Schmitz, J.M., Swann, A.C., 2001. Psychiatric Aspects of Impulsivity. *American Journal of Psychiatry*. 2001, 158(11), pp.1783–1793. https://doi.org/10.1176/appi.ajp.158.11.1783

Morse, A., Benitez, V., Belpaeme, T., Cangelosi, A. and Smith, L., 2015. Posture Affects How Robots and Infants Map Words to Objects. *PLOS ONE,* 10(3). https://doi.org/10.1371/journal.pone.0116012

Patton, J.H., Stanford, M.S., Barratt, E.S., 1995. Factor Structure of the Barratt Impulsiveness Scale. *J Clin Psy*, vol. 51, pp. 768-774. https://doi.org/10.1002/1097-4679(199511)51:6%3C768::aid-jclp2270510607%3E3.0.co;2-1

Pavlov, I.P., 1997. The Work of the Digestive Glands (W.H. Thompson, Trans.). *American Psychologist*, 52(9), pp.936–940. https://doi.org/10.1037/0003-066X.52.9.936

Quinlan, J., 1983. Learning Efficient Classification Procedures and Their Application to Chess End Games. *Machine Learning,* pp.463-482. https://doi.org/10.1007/978-3-662-12405-5_15

Rim, H.B., 2012. Maximizing, Satisficing and Their Impacts on Decision-Making Behaviors. *PhD Thesis. The Ohio State University, OH, USA.*

Rosenblatt, F., 1958. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, 65(6), pp.386-408. https://doi.org/10.1037/h0042519

Samuel, A., 1959. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3(3), pp.210-229. https://doi.org/10.1147/rd.33.0210

Sato, Y., Iida, H. and van den Herik, H., 2015. Transfer Learning by Inductive Logic Programming. *Advances in Computer Games*, pp.223–234. https://doi.org/10.1007/978-3-319-27992-3_20

Schultz, W., Dayan, P. and Montague, P., 1997. A Neural Substrate of Prediction and Reward. *Science,* 275(5306), pp.1593-1599. https://doi.org/10.1126/science.275.5306.1593

Schwartz, B., Ward, A., Monterosso, J., Lyubomirsky, S., White, K. and Lehman, D., 2002. Maximizing versus Satisficing: Happiness is a Matter of Choice. *Journal of Personality and Social Psychology*, 83(5), pp.1178-1197. https://doi.org/10.1037/0022-3514.83.5.1178

Seymour, B., O'Doherty, J.P., Dayan, P., Koltzenburg, M., Jones, A.K., Dolan, R.J., Friston, K.J., Frackowiak, R.S., 2004. Temporal Difference Models Describe Higher-Order Learning in Humans. *Nature*, 10, 429(6992), pp.664-7. https://doi.org/10.1038/nature02581

Silver, D., Huang, A., Maddison, C. *et al.*, 2016. Mastering the Game of Go with Deep Neural Networks and Tree Search. *Nature* 529, pp.484–489. https://doi.org/10.1038/nature16961

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K. and Hassabis, D., 2018. A General Reinforcement Learning Algorithm that Masters Chess, Shogi, and Go Through Self-Play. *Science*, 362(6419), pp.1140-1144. https://doi.org/10.48550/arXiv.1712.01815

Simon, H., 1956. Rational Choice and the Structure of the Environment. *Psychological Review*, 63(2), pp.129-138. https://doi.org/10.1037/h0042769

Sirota, M. and Juanchich, M., 2018. Effect of Response Format on Cognitive Reflection: Validating a Two- and Four-Option Multiple Choice Question Version of the Cognitive Reflection Test. *Behaviour Research Methods*, 50(6), pp.2511-2522. https://doi.org/10.3758/s13428-018-1029-4

Skinner, B., 1938. The Behavior of Organisms. *New York: Appleton-Century-Crofts,* pp.457.

Spunt, R., Rassin, E. and Epstein, L., 2009. Aversive and Avoidant Indecisiveness: Roles for Regret Proneness, Maximization, and BIS/BAS Sensitivities. *Personality and Individual Differences*, 47(4), pp.256-261. https://doi.org/10.1016/j.paid.2009.03.009

Sutton, R. and Barto, A., 2018. Reinforcement Learning. *Massachusetts: MIT Press Ltd, Cambridge, MA, USA.*

Tesauro, G., 1995. Temporal Difference Learning and TD-Gammon. *Communications of the ACM*, 38(3), pp.58-68. https://doi.org/10.1145/203330.203343

Tamatsukuri, A. and Takahashi, T., 2019. Guaranteed Satisficing and Finite Regret: Analysis of a Cognitive Satisficing Value Function. *Biosystems*, 180, pp.46-53. https://doi.org/10.48550/arXiv.1812.05795

Thorndike E.L. 1898. Animal Intelligence: An Experimental Study of the Association Processes in Animals. *The Psychological Review: Monograph Supplements*, 2(4). https://doi.org/10.1037/h0092987

Turing, A., 1950. Computing Machinery and Intelligence. *Mind*, LIX(236), pp.433-460. https://doi.org/10.1093/mind/LIX.236.433

Wang, J., Kurth-Nelson, Z., Kumaran, D., Tirumala, D., Soyer, H., Leibo, J., Hassabis, D. and Botvinick, M., 2018. Prefrontal Cortex as a Meta-Reinforcement Learning System. *Nature Neuroscience*, 21(6), pp.860-868. https://doi.org/10.1038/s41593-018-0147-8

Wason, P. and Evans, J., 1974. Dual Processes in Reasoning? *Cognition*, 3(2), pp.141-154. https://doi.org/10.1016/0010-0277(74)90017-1

Watkins, C., 1989. Learning from Delayed Rewards. *PhD Thesis. King’s College, Cambridge, UK.*

Watson, J., 1913. Psychology as the Behaviourist Views It. *Psychological Review*, 20(2), pp.158-177. https://doi.org/10.1037/h0074428

Weaver, L. and Bossomaier, T., 1998. Evolution of Neural Networks to Play the Game of Dots-and-Boxes. *ArXiv*. https://doi.org/10.48550/arXiv.cs/9809111

Zador, A., 2019. A Critique of Pure Learning and What Artificial Neural Networks can Learn from Animal Brains. *Nature Communications*, 10(1). https://doi.org/10.1038/s41467-019-11786-6

Zhou, W., Chen, Y. and Li, J., 2019. Competitive Evolution Multi-Agent Deep Reinforcement Learning. *Proceedings of the 3rd International Conference on Computer Science and Application Engineering - CSAE 2019*, 24, pp.1-6. https://doi.org/10.1145/3331453.3360975

Supplementary Tables

	Under 18	18-24	25-34	35-44	45-54	55 or older
No. Participants	2	59	3	1	3	3

Table 1: Distribution of all participant ages.

	Female	Male	Non-binary	Prefer not to say
No. Participants	47	15	2	7

Table 2: Distribution of all participant genders.

BIS/BAS/CRT Groups	**Mean Sq**	**F Value**	**Pr**	**Significance**	**DF**
BIS High	3.009	1.468	0.227		1
BIS High:Quartile	0.305	0.149	0.93		3
BAS Reward High	0.632	0.306	0.581		1
BAS Reward High:Quartile	0.141	0.068	0.977		3
BAS Fun High	5.446	2.702	0.102		1
BAS Fun High:Quartile	1.565	0.777	0.508		3
BAS Drive High	26.664	14.035	0.000241	***	1
BAS Drive High:Quartile	1.408	0.741	0.528966		3
CRT Correct High	41.72	22.801	3.71E-06	***	1
CRT Correct High:Quartile	0.59	0.324	0.808		3
CRT Intuitive High	34.54	18.582	2.68E-05	***	1
CRT Intuitive High:Quartile	1.25	0.673	0.569		3
Total					70

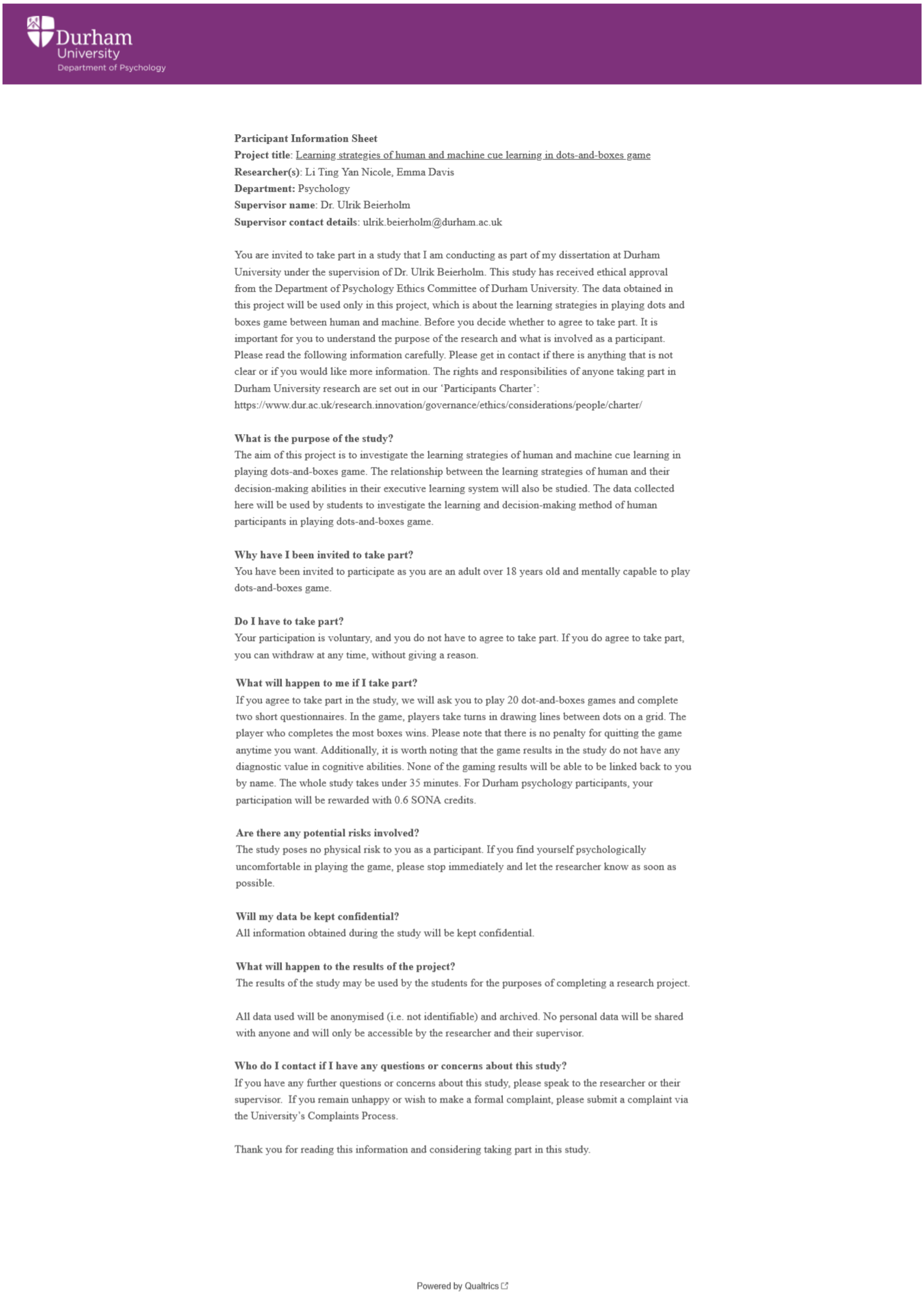
Table 3: Two-way ANOVA test results determining effect of high BIS/BAS/CRT and game quartiles on participant scores. Effects of high scoring in individual BIS/BAS/CRT groups on score and statistical significance (with Pr as p-values) displayed as well as interaction between BIS/BAS/CRT groups and quartile. CRT Correct, CRT Intuitive and BAS Drive effect on game score is statistically significant.

			95% Confidence Interval		
BIS/BAS/CRT Groups	**Quartile Comparisons**	**Mean Difference**	**Lower Bound**	**Upper Bound**	**Pr adj**
BIS High	Q2-Q1	3.19E-01	-0.44669	1.084984	0.701776
	Q3-Q1	5.96E-01	-0.17009	1.36158	0.185461
	Q4-Q1	5.96E-01	-0.17009	1.36158	0.185461
	Q3-Q2	2.77E-01	-0.48924	1.042431	0.785215
	Q4-Q2	2.77E-01	-0.48924	1.042431	0.785215
	Q4-Q3	-4.44E-16	-0.76583	0.765835	1
BAS Drive High	Q2-Q1	3.19E-01	-0.41817	1.056467	0.676168
	Q3-Q1	5.96E-01	-0.14157	1.333063	0.158568
	Q4-Q1	5.96E-01	-0.14157	1.333063	0.158568

	Q3-Q2	2.77E-01	-0.46072	1.013914	0.765096
	Q4-Q2	2.77E-01	-0.46072	1.013914	0.765096
	Q4-Q3	-4.44E-16	-0.73732	0.737318	1
BAS Fun High	Q2-Q1	3.19E-01	-0.44021	1.078504	0.696168
	Q3-Q1	5.96E-01	-0.16361	1.355099	0.179264
	Q4-Q1	5.96E-01	-0.16361	1.355099	0.179264
	Q3-Q2	2.77E-01	-0.48276	1.03595	0.780833
	Q4-Q2	2.77E-01	-0.48276	1.03595	0.780833
	Q4-Q3	4.44E-16	-0.75935	0.759355	1
BAS Reward High	Q2-Q1	3.19E-01	-0.44966	1.087956	0.704308
	Q3-Q1	5.96E-01	-0.17306	1.364551	0.188317
	Q4-Q1	5.96E-01	-0.17306	1.364551	0.188317
	Q3-Q2	2.77E-01	-0.49221	1.045403	0.787189
	Q4-Q2	2.77E-01	-0.49221	1.045403	0.787189
	Q4-Q3	0.00E+00	-0.76881	0.768807	1
CRT Correct High	Q2-Q1	3.19E-01	-0.40445	1.042746	0.662959
	Q3-Q1	5.96E-01	-0.12785	1.319342	0.146032
	Q4-Q1	5.96E-01	-0.12785	1.319342	0.146032
	Q3-Q2	2.77E-01	-0.447	1.000193	0.754607
	Q4-Q2	2.77E-01	-0.447	1.000193	0.754607
	Q4-Q3	1.33E-15	-0.7236	0.723597	1
CRT Intuitive High	Q2-Q1	3.19E-01	-0.41014	1.048439	0.668512
	Q3-Q1	5.96E-01	-0.13355	1.325035	0.151197
	Q4-Q1	5.96E-01	-0.13355	1.325035	0.151197
	Q3-Q2	2.77E-01	-0.45269	1.005886	0.759026
	Q4-Q2	2.77E-01	-0.45269	1.005886	0.759026
	Q4-Q3	4.44E-16	-0.72929	0.72929	1

Table 4: Two-way ANOVA test results after TukeyHSD (Honest Significant Differences) pairwise tests between game quartiles; Q1=games 1-5, Q2=games 6-10, Q3=games 11-15, Q4=games 16-20. Pr adj is p-values after adjustment for multiple comparisons. No pairwise comparisons are statistically significant.

Supplementary Figures



Participant Information Sheet
Project title: Learning strategies of human and machine cue learning in dots-and-boxes game
Researcher(s): Li Ting Yan Nicole, Emma Davis
Department: Psychology
Supervisor name: Dr. Ulrik Beierholm
Supervisor contact details: ulrik.beierholm@durham.ac.uk

You are invited to take part in a study that I am conducting as part of my dissertation at Durham University under the supervision of Dr. Ulrik Beierholm. This study has received ethical approval from the Department of Psychology Ethics Committee of Durham University. The data obtained in this project will be used only in this project, which is about the learning strategies in playing dots and boxes game between human and machine. Before you decide whether to agree to take part. It is important for you to understand the purpose of the research and what is involved as a participant. Please read the following information carefully. Please get in contact if there is anything that is not clear or if you would like more information. The rights and responsibilities of anyone taking part in Durham University research are set out in our 'Participants Charter': https://www.dur.ac.uk/research.innovation/governance/ethics/considerations/people/charter/

What is the purpose of the study?
The aim of this project is to investigate the learning strategies of human and machine cue learning in playing dots-and-boxes game. The relationship between the learning strategies of human and their decision-making abilities in their executive learning system will also be studied. The data collected here will be used by students to investigate the learning and decision-making method of human participants in playing dots-and-boxes game.

Why have I been invited to take part?
You have been invited to participate as you are an adult over 18 years old and mentally capable to play dots-and-boxes game.

Do I have to take part?
Your participation is voluntary, and you do not have to agree to take part. If you do agree to take part, you can withdraw at any time, without giving a reason.

What will happen to me if I take part?
If you agree to take part in the study, we will ask you to play 20 dot-and-boxes games and complete two short questionnaires. In the game, players take turns in drawing lines between dots on a grid. The player who completes the most boxes wins. Please note that there is no penalty for quitting the game anytime you want. Additionally, it is worth noting that the game results in the study do not have any diagnostic value in cognitive abilities. None of the gaming results will be able to be linked back to you by name. The whole study takes under 35 minutes. For Durham psychology participants, your participation will be rewarded with 0.6 SONA credits.

Are there any potential risks involved?
The study poses no physical risk to you as a participant. If you find yourself psychologically uncomfortable in playing the game, please stop immediately and let the researcher know as soon as possible.

Will my data be kept confidential?
All information obtained during the study will be kept confidential.

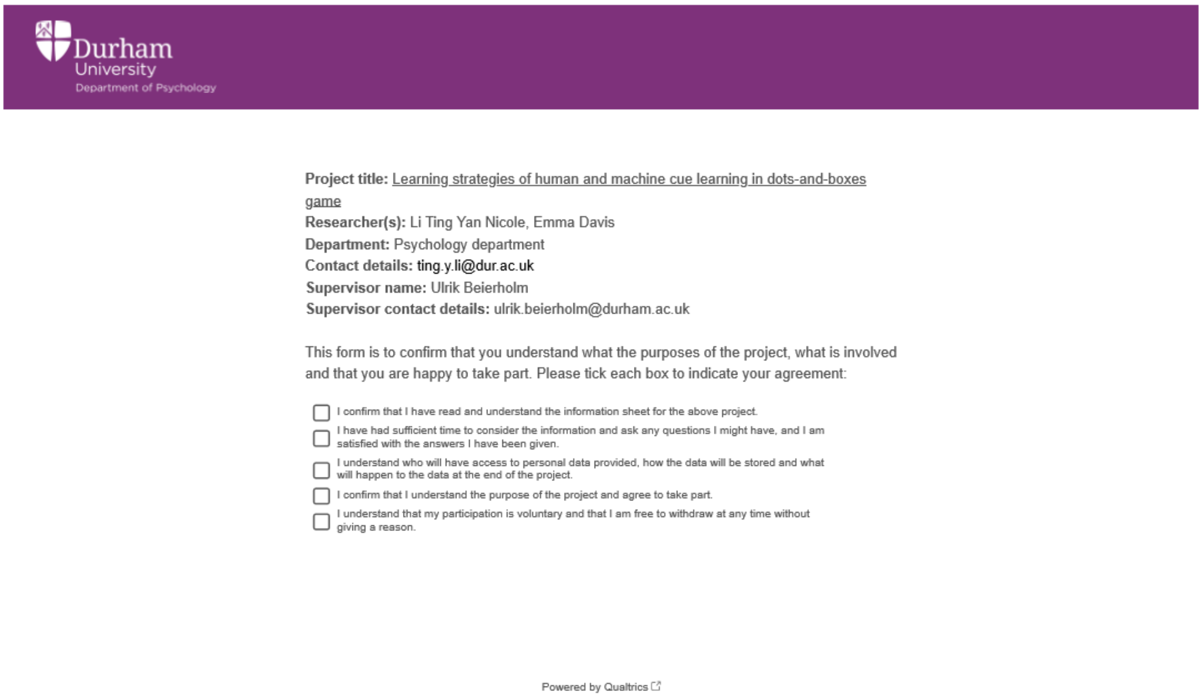
What will happen to the results of the project?
The results of the study may be used by the students for the purposes of completing a research project.

All data used will be anonymised (i.e. not identifiable) and archived. No personal data will be shared with anyone and will only be accessible by the researcher and their supervisor.

Who do I contact if I have any questions or concerns about this study?
If you have any further questions or concerns about this study, please speak to the researcher or their supervisor. If you remain unhappy or wish to make a formal complaint, please submit a complaint via the University's Complaints Process.

Thank you for reading this information and considering taking part in this study.

Supplementary Figure 1: Study information all volunteers are provided with before participating.



Durham University
Department of Psychology

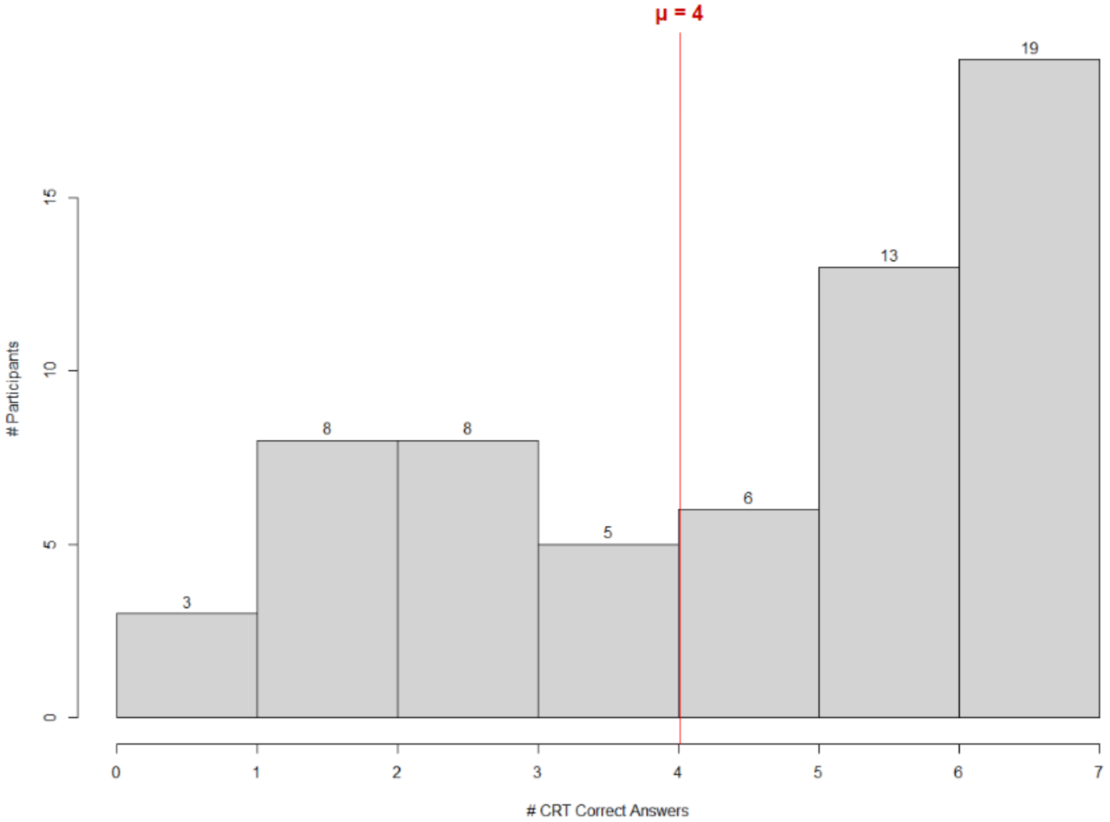
Project title: Learning strategies of human and machine cue learning in dots-and-boxes game
Researcher(s): Li Ting Yan Nicole, Emma Davis
Department: Psychology department
Contact details: ting.y.li@dur.ac.uk
Supervisor name: Ulrik Beierholm
Supervisor contact details: ulrik.beierholm@durham.ac.uk

This form is to confirm that you understand what the purposes of the project, what is involved and that you are happy to take part. Please tick each box to indicate your agreement:

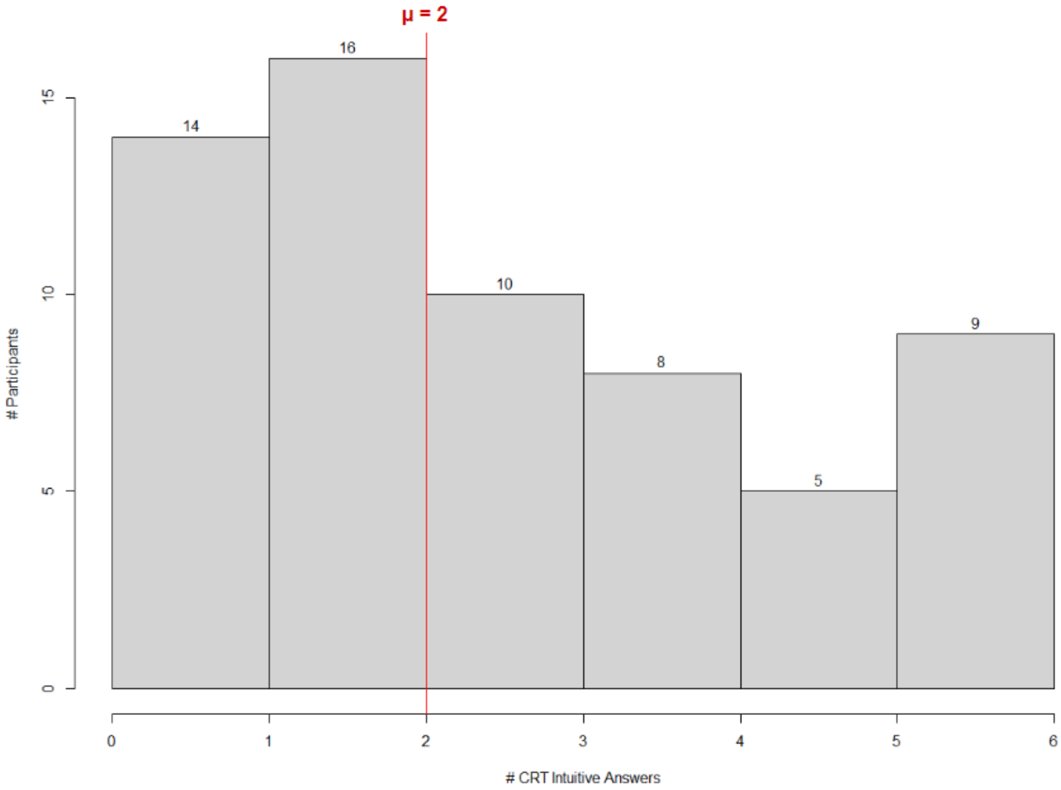
☐ I confirm that I have read and understand the information sheet for the above project.
☐ I have had sufficient time to consider the information and ask any questions I might have, and I am satisfied with the answers I have been given.
☐ I understand who will have access to personal data provided, how the data will be stored and what will happen to the data at the end of the project.
☐ I confirm that I understand the purpose of the project and agree to take part.
☐ I understand that my participation is voluntary and that I am free to withdraw at any time without giving a reason.

Powered by Qualtrics

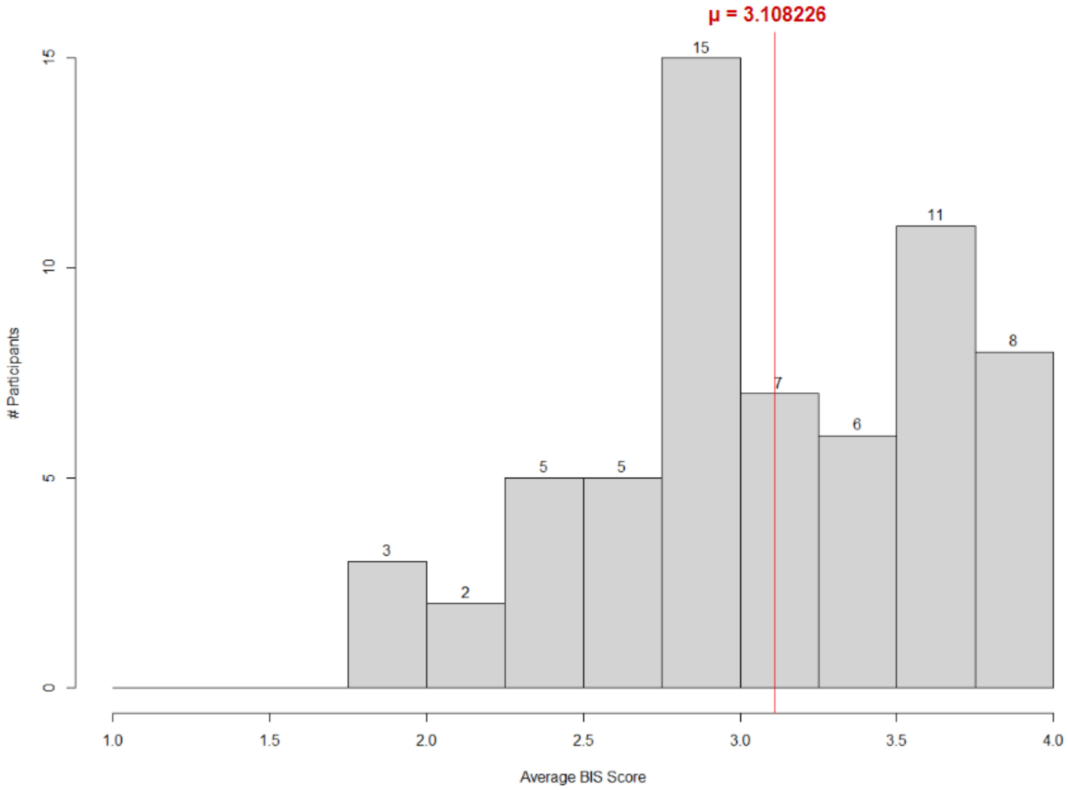
Supplementary Figure 2: Consent form all volunteers must read and sign before participating.



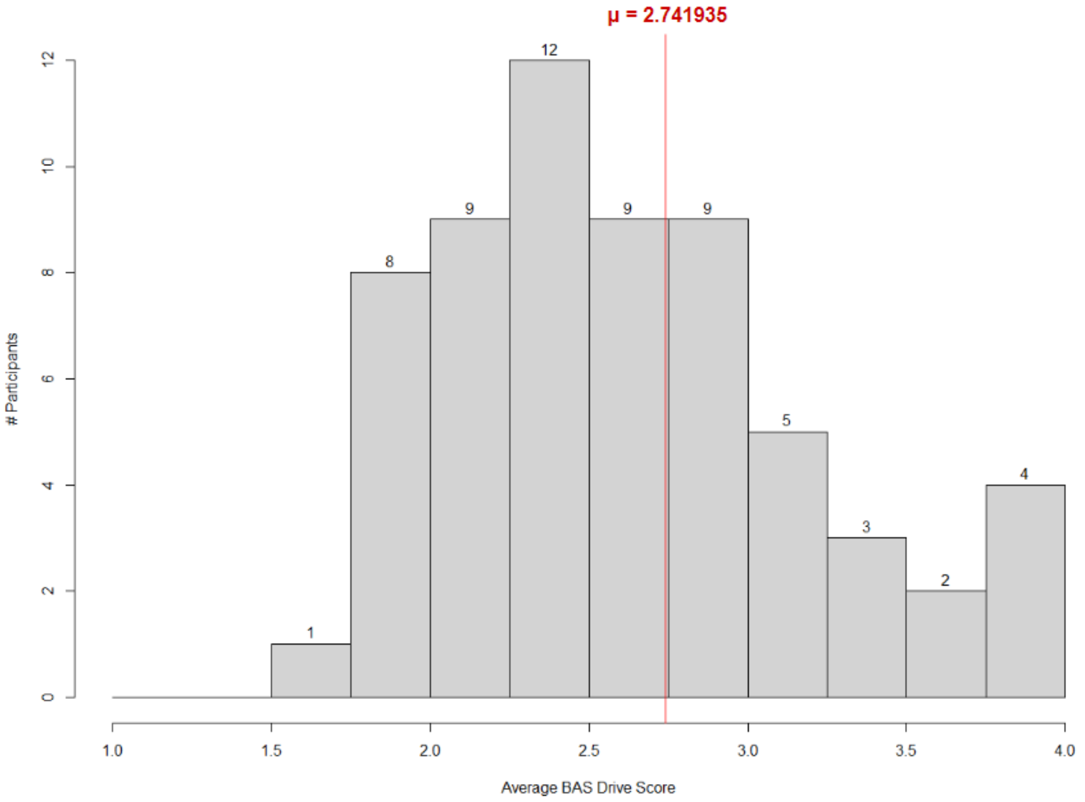
Supplementary Figure 3a: Distribution of CRT correct scores of all 62 volunteers. All volunteers with scores > μ were classed as belonging to group CRT Correct High (n = 38).



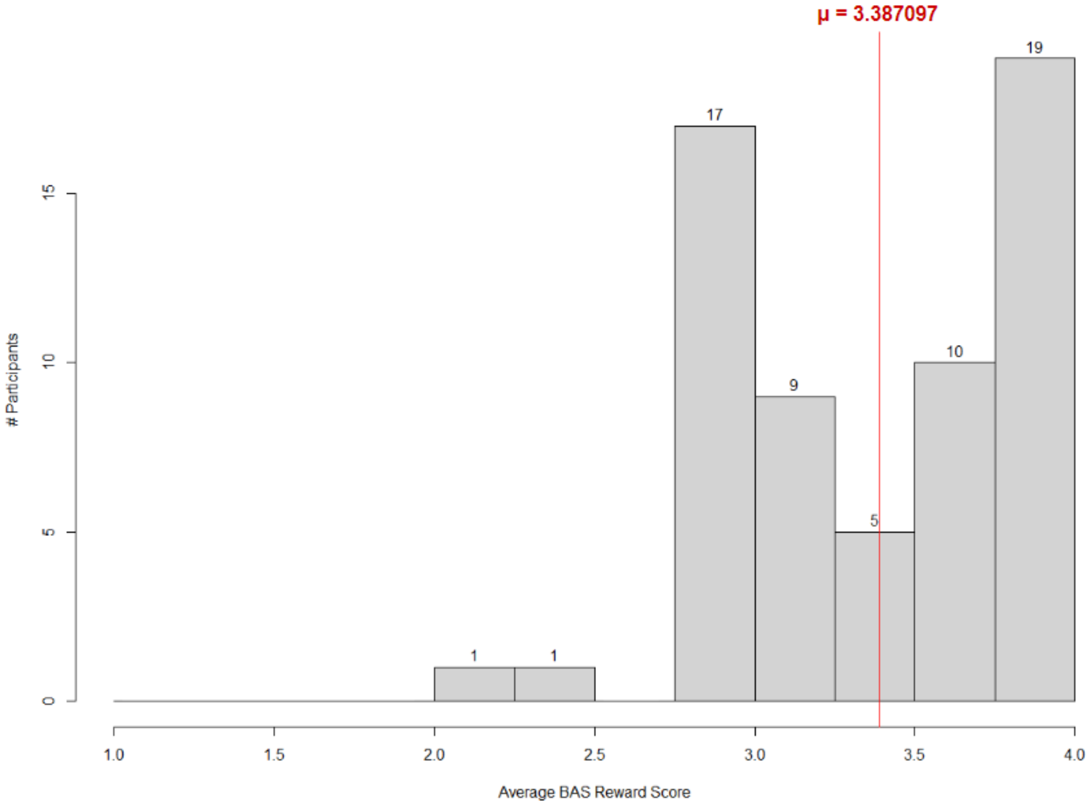
Supplementary Figure 3b: Distribution of CRT intuitive scores of all 62 volunteers. All volunteers with scores > μ were classed as belonging to group CRT Intuitive High (n = 32).



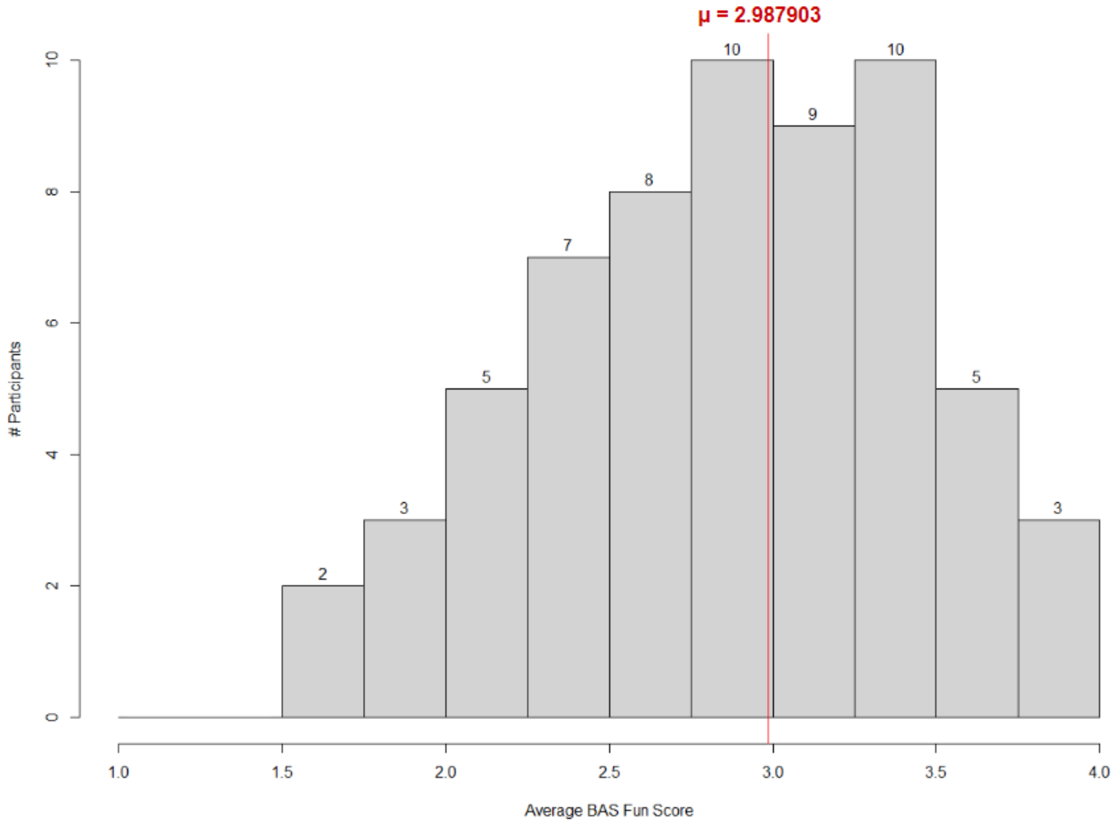
Supplementary Figure 3c: Distribution of the average BIS scores of all 62 volunteers. All volunteers with average scores > μ were classed as belonging to group BIS High (n = 32).



Supplementary Figure 3d: Distribution of the average BAS Drive scores of all 62 volunteers. All volunteers with average scores > μ were classed as belonging to group BAS Drive High (n = 32).



Supplementary Figure 3e: Distribution of the average BAS Reward scores of all 62 volunteers. All volunteers with average scores > μ were classed as belonging to group BAS Reward High (n = 34).



Supplementary Figure 3f: Distribution of the average BAS Fun scores of all 62 volunteers. All volunteers with average scores > μ were classed as belonging to group BAS Fun High (n = 27).